Efficient quantum communication protocols and asynchronism in the toric code

By

João F. Doriguello

School of Mathematics
University of Bristol

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Science.

September 2021

Word count: 34928

Abstract

This thesis studies quantum communication complexity and quantum error correction, two fundamental areas in the theory of quantum computation, and is divided into two parts. The first part focuses on generalising three central problems in quantum communication complexity, often by the introduction of more general Boolean functions. The first studied problem is the one of approximating the Hamming distance in the simultaneous message passing model in communication complexity — a natural generalisation of the Equality problem. An efficient quantum communication protocol is established and later used as the basis for approximating other useful distance measures, e.g. graph distance and ℓ_1-distance.

The second generalised problem is the Boolean Hidden Matching problem, the first proposed one to exhibit an exponential quantum-classical communication separation in the one-way model. Originally defined as a task of matching bits using the Parity function, the problem is generalised by replacing the Parity function with any Boolean function. The hardness of the resulting problem, both classical and quantumly, is characterised in terms of two properties of the Boolean function used, its sign degree and pure high degree.

The first part of the thesis is completed by studying the generalisation of random access codes where the parties are no longer interested in recovering any initial bit, but the value of a fixed Boolean function on any subset of the initial bits with constant size. Different random access codes are defined by means of different accessible classical and quantum resources and their efficiency analysed and compared.

The second part of the thesis is devoted to the study of quantum error correction and the toric code. Even though unrealistic for some physical platforms, most decoders proposed for the toric code assume that stabilizer measurements can always be performed deterministically. This second part aims to study the effect of probabilistic stabilizer measurements, i.e., asynchronism, on the toric code. A few different decoders, based on gradually more refined approximations, are proposed in order to handle the asynchronism and it is numerically shown that they are able to maintain a high threshold even in the limit where stabilizer measurements are performed continuously.

Dedication and acknowledgements

My deepest thanks goes to my supervisor, Ashley Montanaro, who supported and motivated me during these last three years. Coming from a physics background, I am grateful to Ashley for introducing me to the field of Quantum Computation, and also for helping me to mature both academically and intellectually since my Masters. I also thank Ashley for incentivising my taste for literature by lending me his copy of John Milton's Paradise Lost.

I thank my co-authors, Naomi Nickerson and Hugo Cable, for all the help and discussions behind Chapter 6 about quantum error correction and the toric code, and for their kind welcome at the time of my visit at PsiQuantum.

I also thank Ryan Mann for many useful discussions, most of them about concepts from Chapter 6 and self-avoiding walks; Ronald de Wolf for having reliably improved all my papers so far with valuable suggestions and comments, for suggesting the block-encoding scheme in Chapter 4, originally with the $2, 3 \mapsto 1$ QRACs, for pointing out Refs. [119, 121, 177] and for the precise reviewing of the thesis; Noah Linden for the precise reviewing of the thesis and for testing me throughout my PhD with interesting questions; Raphaël Clifford for helpful discussions related to Chapter 2; Makrand Sinha for useful discussions about the hypercontractive inequality; Máté Farkas and Mark Howard for pointing out Refs. [4, 72, 73, 190] and [69], respectively.

These last four years would not be the same without all the support from QE-CDT (Center for Doctoral Training in Quantum Engineering) and QET Labs (Quantum Engineering Technology Labs) and all the people behind them, both the management and the academic teams. And not less important, my time in Bristol would definitely not be the same without my cohort: Brian Flynn, David Ibberson, George Atkinson, Giorgos Eftaxias, Joe Lennon, Jorge Monroy Ruz, Konstantina Koteva, Max Wilson, Rachel Chadwick, Ross Wakefield and Will Dixon.

Finally, I am very grateful to my family and friends from Brazil, whom I could not see as much as I would like to during these last four years, and to all the new friends I made in Bristol.

I acknowledge the use of the computational facilities of the Advanced Computing Research Centre, University of Bristol - http://www.bris.ac.uk/acrc/ - in obtaining the data of Chapter 6.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

DATE: 29/10/2021

Table of Contents

Chapter 0

Introduction

The field of computational complexity and information theory went through impactful changes during the last few decades, starting in the early 1980s with Benioff [25, 26], Feynman [75, 76] and Manin [139], when it was realized that quantum resources can offer great advantages in many areas, e.g. algorithms, communication and cryptography, over their classical counterpart. The first proposed algorithms by Deutsch and Jozsa [59] and by Simon [182] and, later, the breakthrough results of Shor's algorithmfrom 1994 [178, 180] for factorising large numbers and Grover's search algorithm from 1996 [90] showed the existence of efficient algorithms in solving major problems. Cryptographic protocols secured by quantum mechanics, such as the BB84 key distribution protocol [28], were also proposed and fundamental ideas such as Holevo's theorem [100], Schumacher compression [173] and entanglement distillation formed the foundations of an information theory of quantum systems in terms of quantum bits, or qubits. Quantum communication complexity, initially proposed by Yao in 1993 [208], is one small part of this great field and a fine example of the transformations promoted by quantum concepts. On the other hand, quantum error-correction and fault-tolerant quantum computation, discovered independently by Shor in 1995 [179] and Steane in 1996 [185], demonstrated the experimental feasibility of all these ideas.

This thesis is divided in two parts. In the first part, we explore a few generalisations of famous results pertaining to quantum communication complexity, while in the second part we explore a particular problem in quantum error-correction. The field of communication complexity is particularly interesting in that one can prove *unconditional* advantages, i.e., communication savings, with the use of quantum resources over classical resources, in opposition to computational complexity. Moreover, due to its simplicity and generality, the ideas of communication complexity often find application into many other, sometimes seemingly unrelated, areas.

Part I begins with Chapter 1, which reviews the basic concepts from classical and quantum communication complexity and the analysis of Boolean functions that shall be important throughout Part I. It then continues on to Chapters 2, 3 and 4, which contain original work and are the main focus of the thesis. In Chapter 2 we study the quantum communication complexity of the problem of approximating the Hamming distance up to relative error in the Simultaneous Message Passing (SMP) model. In this model, the two parties are only allowed to communicate a single message to a third party, who outputs the answer. The considered problem is a natural generalisation of computing the Equality function, which was the first problem to present an exponential separation between classical and quantum communication complexities in the SMP model [39]. Classically, the Equality function, and thus approximating the Hamming distance up to relative error, is known to require $\Omega(\sqrt{n})$ bits of communication in such a model. Here we present a protocol that solves our Hamming distance approximation problem by communicating $\tilde{O}(\log^2 n)$ qubits (where the notation $\tilde{O}$ hides polyloglog factors). We then explore such a protocol for approximating the distance between vertices in a graph which can be isometrically embedded into the hypercube.

Chapter 3 proposes a very broad generalisation of the Boolean Hidden Matching (BHM) problem, which was the first problem [81] to exhibit an exponential separation between classical and quantum communication complexities in the one-way model (only one party is allowed to communicate). In this rather intricate problem, one party receives a string x, while the other receives a second string w and a matching of the bits from x. By computing each parity of the matched bits, the second party obtains a 'compressed' version of x. It is then promised that such a compressed version is either equal to w or its complement, the question thus being to decide on the correct case. In practice, the second party needs to obtain only one set of bits of x that are matched together. Such task requires $\Omega(\sqrt{n})$ classical bits, but can be solved using $\log n$ qubits. Our contribution is to replace the Parity function when 'compressing' the string x with any Boolean function f. The resulting problem is named *f-Boolean Hidden Partition problem*. We then partially characterize the hardness of the problem by the *sign-degree* of f, which is the smallest degree of a polynomial $p(x)$ such that $f(x) = \mathrm{sgn}(p(x))$. The hardness of the problem is characterized by the sign-degree of f. We present an efficient classical communication protocol when the sign-degree is less than or equal to 1, and an efficient quantum communication protocol when it is less than or equal to 2. The classical hardness of all symmetric functions of sign-degree greater than or equal to 2 is also characterized, except for one family of specific cases. Finally, via Fourier analysis, we also prove a classical lower bound for any function whose pure high degree (the highest degree of its non-zero Fourier coefficients) is greater than or equal to 2, and a quantum lower bound for pure high degree greater than or equal to 3. While these results give a large family of new exponential classical-quantum communication separations, for some Boolean functions the complexity of the problem is left undecided, since the pure high degree of a function is always less than or equal to its sign-degree.

Chapter 4 explores the concept of quantum random access (QRAC) codes [12], which is an encoding of bits into a smaller number of qubits, such that, any one of the initial bits can be recovered with high probability of success. Even though we do not explicitly frame the problem in a communication complexity setting, QRACs are equivalent to computing the *Index function* in the one-way model. In this chapter we generalise the idea of (quantum) random access codes to recovering not just an initial bit, but the value of a fixed Boolean function on any subset of the initial bits with fixed size. We call them *f-random access codes*. The case of the Parity function was already considered in [24], and here we generalise to arbitrary Boolean functions. We propose a series of f-random access codes depending the resources employed: classical (f-RAC) and quantum (f-QRAC) encoding, together with many different resources, e.g. private or shared randomness, shared entanglement (f-entanglement assisted random access code, or simply f-EARAC) and Popescu-Rohrlich boxes (f-Popescu-Rohrlich random access code, or simply f-PRRAC). The success probability of our f-random access codes is characterized by the *noise stability* of f, which is the correlation between $f(x)$ and $f(y)$ when y is obtained from x by independently flipping its bits with some probability. We show that quantum resources offer a limited advantage over classical ones. This point is strengthened by our upper bound on the success probability of any f-QRAC which matches our success probability lower bounds up to multiplicative constants. We conjecture that such an upper bound could be extended to the stronger case of f-EARACs. On the other hand, we show that the use of stronger-than-quantum correlations like Popescu-Rohrlich boxes can lead to extremely powerful f-random access codes.

Part II contains Chapters 5 and 6, and deals with a very different topic than communication complexity: quantum error-correction. In Chapter 5 we review the basics of the stabilizer formalism and the toric code introduced by Kitaev [118]. In Chapter 6 we present our original work on the toric code. It is often assumed that stabilizer operators are either perfect or have some small probability of returning a wrong result, and that they can be measured deterministically, which is a reasonable assumption for many systems. However, for some systems e.g. linear-optical quantum computing, stabilizer measurements are inherently probabilistic. For all decoders (the algorithm that identifies the correction operator) presented up to now, it has been assumed that stabilizer measurements can always be obtained at once — a situation we call *sychronous* stabilizer measurements. In this chapter we study the effect of *asychronous* stabilizer measurements on the toric code. We present a simple model of asynchronism and modify the very well known Minimum Weight Perfect Matching (MWPM) decoder to deal with such asynchronism. In this model the stabilizer measurements are attempted at discrete times and each attempt provides a parity outcome with probability s, called the *synchronicity* parameter. In the limit $s \to 0$ the outcomes of stabilizer measurements are received at completely random times. The usual MWPM decoder, which returns the most likely (with less energy) possible error configuration as the correction operator, is modified by using an edge-contraction idea from [184]. Under the error model of independent and identical X and Z Pauli errors, the

resulting decoder presents a high threshold value in terms of the physical error on the qubits for all values of synchronicity — in particular, under a completely continuous model of syndrome extraction it can maintain a threshold of 1.688%, which is relatively close to 2.93% [193] for fully deterministic measurements. We also study the effect of degeneracy in our decoder, an idea that was considered before for the case of perfect measurements in the toric code and which was shown to improve the threshold values considerably [184]. We define a series of decoders based on our main MWPM decoder which account for increasing high-order degeneracy. We then numerically show that the threshold improvement provided by high-order degeneracy is substantially reduced with asynchronism, thus rendering the use of degeneracy unnecessary in some physical platforms.

Previous Publications

Most of this thesis has been published previously.

- Chapter 2 is joint work with Ashley Montanaro and has been published previously as "Quantum sketching protocols for Hamming distance and beyond", *Physical Review A* 99.6 (2019): 062331 (`arxiv:1810.12808`).
- Chapter 3 is joint work with Ashley Montanaro and has been published previously as "Exponential Quantum Communication Reductions from Generalizations of the Boolean Hidden Matching Problem", *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020 (`arxiv:2001.05553`).
- Chapter 4 is joint work with Ashley Montanaro and has been published as "Quantum random access codes for Boolean functions", *Quantum* 5, 402 (2021) (`arxiv:2011.06535`).
- Chapter 6 is joint work with Hugo Cable and Naomi Nickerson and is being prepared for publication.

Part I

Chapter 1

Communication Complexity and Boolean Functions

In this chapter we briefly review the area of classical and quantum communication complexity (Section 1.1) and define some basic tools in analysis of Boolean functions (Section 1.2) used throughout Part I. Two introductions to classical communication complexity are the books from Kushilevitz and Nisan [129] and from Rao and Yehudayoff [170]. We stick to the notation from [129]. An introdution to quantum communication complexity is the paper from Buhrman, Cleve, Massar and de Wolf [38]. Regarding analysis of Boolean functions, most of the reviewed material can be found in O'Donnell's book [157]. See also de Wolf's introduction paper [201].

1.1 Communication Complexity

In 1979, Yao [206] introduced the two-party classical communication complexity model. In this model, two parties (commonly referred to as Alice and Bob) hold different parts of the input information and are asked to solve a task depending on this information. In order to do so, they will need to communicate with each other until the answer to the task can be outputted. The main concern of the model is to perform the task with the minimum amount of communication possible. In other words, Yao assumed the following:

- There are only two communicating parties in the system.
- Each party receives a fixed part of the input information.
- The objective is to compute a pre-determined function of their inputs.
- The only quantity of concern is communication.

Despite its simplicity, this model offers a rich structure where many complex quantities can be defined and many interesting connections be made.

1.1.1 Deterministic Communication Complexity

Let X, Y, Z be arbitrary finite sets and let $f : X \times Y \to Z$ be an arbitrary function. Alice receives $x \in X$ and Bob receives $y \in Y$ and they want to compute $f(x, y)$. In order to correctly determine the output, they will need to communicate according to a fixed *communication protocol* $\mathcal{P}$ depending solely on f. On a high level, a communication protocol is an algorithm that specifies how the communication should be carried out: at each stage, it determines whether the communication terminates or not. If it terminates, then the protocol specifies the output, i.e., $f(x, y)$. If it does not terminate, then the protocol specifies which party sends the next bit of communication. This information must depend solely on past communication (messages), since it is the only knowledge common to both parties, and also on the information of the party who is communicating. If Alice is speaking, then her message depends on previous communication and on $x \in X$. If Bob is speaking, then his message depends on previous communication and on $y \in Y$.

We are interested mainly in the communication between the parties, and not in the computations carried out privately, therefore, we assume that Alice and Bob have infinite computational power. The *cost* of the protocol $\mathcal{P}$ on input (x, y) is the minimum number of bits exchanged by $\mathcal{P}$ on input (x, y). The cost of $\mathcal{P}$ is the worst case (i.e., maximum) cost of $\mathcal{P}$ over all inputs (x, y). The *deterministic communication complexity* of f is the minimum cost over all protocols $\mathcal{P}$ that compute f.

We can formalize the above notions in the following statements.

Definition 1.1. A *communication protocol* $\mathcal{P}$ over the domain $X \times Y$ with range Z is a rooted binary tree where each internal node is owned either by Alice or Bob. Internal nodes v owned by Alice are labelled by a function $a_v : X \to \{-1, 1\}$ and internal nodes v owned by Bob are labelled by $b_v : Y \to \{-1, 1\}$. The leaves of the binary tree are labelled by $z \in Z$.

The value of $\mathcal{P}$ on input (x, y) is the leaf reached by following the tree starting at the root. At each internal node v the output of a_v, b_v determines which direction to walk: left if -1 and right if 1. The cost of $\mathcal{P}$ on input (x, y) is the length of the walked path on (x, y). The cost of $\mathcal{P}$ is the height/depth of the tree.

Definition 1.2. Given a function $f : X \times Y \to Z$, its *deterministic communication complexity* $D(f)$ is the minimum cost of $\mathcal{P}$ over all protocols $\mathcal{P}$ that compute f.

The deterministic communication complexity definition (and the ones that will follow) can be extended to functions $f : \mathcal{D} \to Z$ where $\mathcal{D} \subset X \times Y$, which are called *partial* functions (in contrast to *total* functions where $\mathcal{D} = X \times Y$). For simplicity we shall refer to total functions in what follows, but all complexity measures can also be defined for partial functions.

The above communication setting can be defined slightly differently depending on the context. Sometimes it is required that both parties learn the output $f(x, y)$, while other times it is required that just one party does so. A trivial protocol for any function is to have Alice

sending her whole input $x \in X$ to Bob, who computes $f(x, y)$ privately and then sends the answer back to Alice. Since Alice's input and the answer can be described by $\lceil \log_2 |X| \rceil$ and $\lceil \log_2 |Z| \rceil$ bits, respectively, we thus have the trivial proposition.

Proposition 1.3. *For any function* $f : X \times Y \to Z$, $D(f) \leq \lceil \log_2 |X| \rceil + \lceil \log_2 |Z| \rceil$.

We shall be concerned with functions of the form $f : \{-1, 1\}^n \times \{-1, 1\}^n \to \{-1, 1\}$, so the question of having both or one party learning the answer amounts to a difference of only one bit in the communication complexity. Therefore, we assume that just one party, normally Bob, needs to learn $f(x, y)$.

1.1.2 Randomized Communication Complexity

One of the many ways to expand the deterministic model of communication complexity is by introducing randomness. Now Alice and Bob are allowed to toss coins and to use the outcomes when deciding what messages to send. More formally, Alice has access to a random string r_A of arbitrary length, and similarly Bob has access to a random string r_B. Both strings are picked independently according to some probability distribution. When looking at the protocol tree, internal nodes now depend on the parties' random strings together with previous messages and their inputs, e.g. an internal node owned by Alice is labelled by a function of r_A and x. As before, traversing the protocol tree by starting at the root on input (x, y) leads to a leaf labelled z which defines the output of the protocol, but the reached leaf will depend on the random strings. It is possible that different choices of r_A, r_B lead to different outputs, therefore protocols might err when randomness is allowed.

Definition 1.4. Let $\mathscr{P}$ be a randomized protocol. We say that $\mathscr{P}$ *computes* f *with* ε*-error* if, for every (x, y),

$$\Pr[\mathscr{P}(x, y) = f(x, y)] \geq 1 - \varepsilon, \tag{1.1}$$

where the probability is taken oven the random choices of r_A and r_B.

Such ε-error randomized protocols are sometimes called *Monte-Carlo* protocols. There are other kinds of protocols, e.g. zero error (sometimes called *Las-Vegas* protocols) and one-sided error protocols, but we shall focus on the (two-sided) error protocols from Definition 1.4.

As mentioned before, different choices of r_A, r_B can lead to different outcomes, and thus to different numbers of exchanged bits. A natural way to deal with this is to consider the *worst case* over all r_A, r_B for a given fixed input.

Definition 1.5. The *worst case running time* of a randomized protocol $\mathscr{P}$ on input (x, y) is the maximum number of communicated bits over all the choices of the random strings r_A and r_B. The *worst case cost* of a randomized protocol $\mathscr{P}$ is the maximum worst case running time of $\mathscr{P}$ over all inputs (x, y).

Definition 1.6. Let $0 \leq \varepsilon < 1/2$. Given a function $f : X \times Y \to Z$, its *randomized communication complexity* $R_\varepsilon(f)$ is the minimum worst case cost of $\mathscr{P}$ over all randomized protocols $\mathscr{P}$ that compute f with ε-error. We write $R(f) := R_{1/3}(f)$.

A natural aim of the field is to relate different communication complexity measures. It should be clear that $R_\varepsilon(f) \leq D(f)$ for any $0 \leq \varepsilon < 1/2$: any randomized protocol can simulate a deterministic one by using empty random strings. How much smaller can $R_\varepsilon(f)$ be compared to $D(f)$? The next lemma, stated without proof, shows that it can be at most exponentially smaller.

Lemma 1.7 ([129, Lemma 3.8])**.** $R(f) = \Omega(\log D(f))$.

This bound is tight. One can prove that for the equality function $\mathrm{EQ} : \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$ defined as $\mathrm{EQ}(x,y) = 1$ if $x = y$ and $\mathrm{EQ}(x,y) = -1$ if $x \neq y$, its deterministic complexity is $D(\mathrm{EQ}) = n$ (see [129, Example 1.21] and [170, Theorem 1.16]), while its randomized complexity is $R(\mathrm{EQ}) = O(\log n)$ [129, Example 3.5].

1.1.3 Public Randomness

The random strings held by Alice and Bob were previously considered to be private, but it is also possible to consider public random strings. Both parties have access to a common random string r chosen according to some probability distribution Π. In regard to the protocol tree, internal nodes owned by Alice are now labelled by functions of r and x, while nodes owned by Bob are labelled by functions of r and y. Equivalently, this can be viewed as a distribution $\{\mathscr{P}_r\}_{r \in \Pi}$ of deterministic protocols, i.e., Alice and Bob randomly choose a common string r according to Π and follow the deterministic protocol $\mathscr{P}_r$.

Definition 1.8. A (randomized) public-coin protocol is a probability distribution over deterministic protocols. Its success probability on input (x,y) is the probability of choosing a deterministic protocol, according to some distribution Π, that correctly computes $f(x,y)$. The *(randomized) public-coin communication complexity* $R^{pub}_\varepsilon(f)$ of a function $f : X \times Y \to Z$ is the minimum cost over all public-coin protocols that compute f with ε-error.

Clearly $R^{pub}_\varepsilon(f) \leq R(f)$, since every public-coin protocol can simulate a private coin protocol: the public string r is the concatenation of the private random strings r_A, r_B. More interestingly, every private coin protocol can also simulate a public-coin protocol with a small penalty in the error and in the communication complexity, a result due to Newman [152] (see also [129, Theorem 3.14] and [170, Theorem 3.5]).

Theorem 1.9 ([152])**.** *Let $f : \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$. For every $\delta > 0$ and every $\varepsilon > 0$, $R_{\varepsilon+\delta}(f) \leq R^{pub}_\varepsilon(f) + O(\log(n/\delta))$.*

1.1.4 Distributional Complexity

Previously we considered randomized protocols where a probability distribution is taken over the random choices by the parties and we considered only worst case inputs. In this section we revise a different model, called distributional communication complexity, where the probability distribution is taken over the *inputs*.

Definition 1.10. Let μ be a probability distribution over $X \times Y$. Given $f : X \times Y \to Z$, its (μ, ε)-*distributional communication complexity* $D^{\mu}_{\varepsilon}(f)$ is the minimum cost of deterministic protocols that correctly output $f(x, y)$ on at least a $1 - \varepsilon$ fraction of all inputs of $X \times Y$, weighted by μ.

It is not hard to see that $R^{pub}_{\varepsilon}(f) \geq D^{\mu}_{\varepsilon}(f)$ for any distribution μ, since $R^{pub}_{\varepsilon}(f)$ is measured by the worst case input. It turns out that such bound completely characterizes the randomized public-coin complexity, a result known as Yao's minimax principle [205, 207] (see also [129, Theorem 3.20] and [170, Theorem 3.3]). It follows from von Neumann's minimax theorem of game theory [151].

Theorem 1.11 ([207])**.** $R^{pub}_{\varepsilon}(f) = \max_{\mu} D^{\mu}_{\varepsilon}(f)$.

Theorem 1.11 gives a 'recipe' to obtain lower bounds on $R^{pub}_{\varepsilon}(f)$: one needs to choose a 'hard distribution' μ and prove that the distributional complexity $D^{\mu}_{\varepsilon}(f)$ over this hard distribution is large, from which it follows that the public-coin complexity is also large. Such approach is convenient because it shifts the randomness from the parties' choices to the inputs, which is normally much easier to analyse. We shall use this method in Chapter 3 to prove a lower bound on the public-coin complexity of our generalisation of the Boolean Hidden Matching problem.

1.1.5 Quantum Communication Complexity

So far we have dealt only with classical communication, i.e., Alice and Bob exchange bits, but this requirement can be expanded to include quantum resources. This model was introduced in 1993 again by Yao [208]. Here both parties hold a quantum computer and are allowed to send each other qubits. More formally, a quantum protocol can be defined as follows. The whole system consists of three parts: Alice's private space, the communication channel and Bob's private space. Ignoring workspace qubits from Alice and Bob, which are initially $|0\rangle$, the system starts in the state $|x\rangle|0\rangle|y\rangle$: Alice gets x, the channel is initially empty, and Bob gets y. Alice and Bob, in turns, apply a unitary transformation to their private space and the channel. This amounts to doing their private computations and also putting a message in the channel, i.e., sending qubits to the other party (the *length* of the message equals the number of qubits sent). To rigorously define the model, we restrict the number of qubits sent in each round to a fixed number agreed beforehand, so that the party that communicates next is well defined (although such constraint is rarely considered when devising quantum communication

protocols). At the end of the protocol both parties make a measurement to determine the output. Similarly to the deterministic and randomized communication complexity, we can define a quantum communication complexity.

Definition 1.12. Given $f : X \times Y \to Z$, its *quantum communication complexity* $Q_\varepsilon(f)$ is the minimum communication cost of a quantum protocol that computes $f(x, y)$ with ε-error.

From the definitions it should not be hard to conclude that $Q_\varepsilon(f) \leq R_\varepsilon(f) \leq D(f)$. One of the main interests of the field is to find large separations between quantum and classical communication complexities, ideally exponential separations. At first it might seem that no large savings can occur given Holevo's theorem [100] that no more than n bits of expected classical information can be transmitted by n qubits without entanglement, but this argument turns out to be false. Alice and Bob are not interested in the information contained in n bits, but rather in $f(x, y)$, which is just one bit. The first impressive separation was proved by Buhrman, Cleve and Wigderson [40] in 1998, who used distributed versions of known quantum algorithms such as the Deutsch-Jozsa [59] and Grover [90] algorithms. Since then a few other examples were found. In Chapters 2 and 3 we shall study two problems that show an exponential separation between classical and quantum communications: computing the Hamming distance and the Boolean Hidden Matching problem.

1.1.6 One-Way Communication Complexity

The protocols mentioned until now are *two-way* (or *interactive*), meaning that Alice and Bob take as many turns as necessary sending messages to each other. By introducing restrictions to this model of communication we can still arrive at interesting situations and complexities. One of such examples is by demanding that Alice communicates a single message to Bob, who is not allowed to communicate back and must just output the answer. We call these protocols *one-way*.

Definition 1.13. A *one-way protocol* is a protocol where Alice sends a single message to Bob, who then outputs the answer. Given $f : X \times Y \to Z$, its *one-way deterministic communication complexity* $D^1(f)$ is the minimum cost over all one-way protocols that compute $f(x, y)$ exactly. Other one-way communication complexities like $R^1_\varepsilon(f)$, $R^{1,pub}_\varepsilon(f)$ and $Q^1_\varepsilon(f)$ are defined similarly.

This limited interaction increases the amount of communication required to compute a function (or at least maintains it), so $D(f) \leq D^1(f)$, and similarly for the other complexity quantities. One can prove that, for deterministic and randomized complexities, such separation can be at most exponential.

Lemma 1.14 ([129, Exercise 4.21])**.** *$D(f) \geq \log_2 D^1(f)$ and $R(f) \geq \log_2 R^1(f)$.*

The first quantum-classical exponential separation in the one-way model was shown in 2004 by Bar-Yossef, Jayram and Kerenidis [20] for a relation and later in 2007 for a (partial) function by Gavinsky, Kempe, Kerenidis, Raz and de Wolf [81]. These problems, called Hidden Matching and Boolean Hidden Matching problems, will be explored in details in Chapter 3.

1.1.7 Simultaneous Communication Complexity

A more stringent constraint can be placed on the two-way communication model: that Alice and Bob *simultaneously* each send a single message (depending on their inputs) to a third party, called the *referee*, who must output the answer. This communication model is normally refereed to as the *simultaneous message passing* (SMP) model and was first considered by Yao [206] and later by Kremer, Nisan and Ron [126].

Definition 1.15. A *simultaneous protocol* is a protocol where Alice and Bob send each a single message to a third party called the referee, who then outputs the answer. Given $f : X \times Y \to Z$, its *simultaneous deterministic communication complexity* $D^{\|}(f)$ is the minimum cost over all simultaneous protocols that compute $f(x, y)$ exactly. Other simultaneous communication complexities like $R^{\|}_{\varepsilon}(f), R^{\|,pub}_{\varepsilon}(f)$ (the random string is shared only between Alice and Bob) and $Q^{\|}_{\varepsilon}(f)$ are defined similarly.

The SMP model is the weakest amongst all the three communication models seen so far, meaning that $D(f) \leq D^1(f) \leq D^{\|}(f)$, and similarly for the other complexity quantities. A useful relation between $D^{\|}(f)$ and $R^{\|}(f)$ was proved by Babai and Kimmel [16] (see also [129, Exercise 4.22]).

Theorem 1.16 ([16])**.** *For any* $f : X \times Y \to \{-1, 1\}$, $D^{\|}(f) = O(R^{\|}(f)^2)$.

A straightforward consequence of the above theorem is that $R^{\|}(\mathrm{EQ}) = \Omega(\sqrt{n})$ (remember that $D^{\|}(\mathrm{EQ}) \geq D(\mathrm{EQ}) = n$). This result regarding Equality had been previously obtained by Newman and Szegedy [153], and around the same time Ambainis gave a matching upper bound [8]. In comparison, if Alice and Bob are allowed access to a shared random bit-string, this complexity drops to $R^{\|,pub}(\mathrm{EQ}) = O(1)$ [16].

For completeness, we sketch Ambainis' protocol [8]. Alice and Bob agree beforehand on a suitable error-correcting code and use it to encode their strings. Alice rearranges the bits of her codeword in a square matrix, chooses a row i uniformly at random and sends its label and content to the referee. Bob does the same, but picks a column j instead of a row. The referee compares Alice's jth bit with Bob's ith bit. If they are different, then $x \neq y$ is announced, otherwise, with probability p the referee announces that $x = y$. By tweaking p, the outputted answer is correct with high probability. Note that $O(\sqrt{n})$ bits are communicated.

1.1.7.1 Quantum Fingerprinting

In the SMP model, a natural strategy for computing $f(x, y)$ is for each of Alice and Bob to compress their data to some kind of "sketch" [74, 19], and send the sketches to the referee, who uses them to determine the distance between the corresponding original data sets. As just mentioned, even for one of the simplest distance measures possible – testing equality of n-bit strings – and even if Alice and Bob are allowed a small probability of failure, this task requires $\Theta(\sqrt{n})$ bits of classical communication to the referee. Remarkably, the use of quantum information allows an exponential reduction in the complexity of equality-testing. If Alice and Bob encode their n-bit strings as particular quantum states called *quantum fingerprints*, then it was shown by Buhrman, Cleve, Watrous and de Wolf in 2001 [39] that there exists a quantum protocol that communicates only $O(\log n)$ qubits and succeeds with constant probability arbitrarily close to 1.

The concept of quantum fingerprinting is going to be central in Chapter 2, therefore, we shall overview the protocol used in [39], starting with the concept of quantum fingerprinting.

Definition 1.17. Given $x \in \{-1, 1\}^n$ and an error-correcting code $E : \{-1, 1\}^n \to \{-1, 1\}^m$, the *quantum fingerprint* of x is the $(\lceil \log_2 m \rceil + 1)$-qubit state

$$|h_x\rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} |i\rangle |E(x)_i\rangle, \tag{1.2}$$

where $E(x)_i$ is the i-th bit of $E(x)$.

Equality can be checked by the following procedure. First assume that for fixed $c > 1$ and $0 < \delta < 1$, we have an error-correcting code $E : \{-1, 1\}^n \to \{-1, 1\}^m$ with $m = cn$ such that the distance between different code words $E(x)$ and $E(y)$ is at least $(1 - \delta)m$. Examples of such error-correcting codes are random linear codes and Justesen codes where, for any $c > 2$ and sufficiently large n, $\delta < 9/10 + 1/(15c)$ [111].

Assuming the existence of such an error-correcting code, Alice and Bob first encode their strings into $E(x)$ and $E(y)$, and then send the quantum fingerprints $|h_x\rangle$ and $|h_y\rangle$ to the referee. Note that, if two codewords are equal in at most δm positions, then $\langle h_x | h_y \rangle \leq \delta m/m = \delta$. The referee must be able to distinguish the case where the two states received are equal from the case where their inner product is at most δ in absolute value. This can be accomplished with one-sided error probability by a SWAP test, which is a procedure that measures and outputs the first qubit of the state

$$(H \otimes I)(\text{c-SWAP})(H \otimes I)|0\rangle|h_x\rangle|h_y\rangle. \tag{1.3}$$

Here H is the Hadamard gate, defined by the map $|b\rangle \to \frac{1}{\sqrt{2}}(|0\rangle + (-1)^b|1\rangle)$, SWAP is the operation $|\phi\rangle|\psi\rangle \to |\psi\rangle|\phi\rangle$, and c-SWAP is the controlled-SWAP (controlled by the first qubit).

The result of Eq. (1.3) is the state

$$\frac{1}{2}|0\rangle\big(|h_x\rangle|h_y\rangle + |h_y\rangle|h_x\rangle\big) + \frac{1}{2}|1\rangle\big(|h_x\rangle|h_y\rangle - |h_y\rangle|h_x\rangle\big). \tag{1.4}$$

Measuring the first qubit returns 1 with probability $\frac{1}{2}(1 - |\langle h_x|h_y\rangle|^2)$. This probability is 0 if $x = y$ and is at least $(1-\delta^2)/2$ if $x \neq y$. Hence the procedure determines if $x = y$ with one-sided error probability $(1+\delta^2)/2$. By repeating the test $O(\log(1/\varepsilon))$ times, the error probability can be reduced to any $\varepsilon > 0$. The final number of communicated qubits is $O(\log(n)\log(1/\varepsilon))$.

The above result was later generalised by Yao [209] and Gavinsky, Kempe and de Wolf [84].

Theorem 1.18 ([209, Theorem 1])**.** *Let $f : \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$. If $R^{\|,pub}(f) = O(1)$, then $Q^{\|}(f) = O(\log n)$.*

1.2 Boolean Functions

A Boolean function is a function of the form $f : \{-1,1\}^n \to \{-1,1\}$. It maps n-bit vectors into single bits. We have dealt implicitly with Boolean functions when reviewing communication complexity. In this section we shall revise some results regarding the *analysis of Boolean functions*, which refers to studying Boolean functions from their Fourier expansion and other analytic means. This topic has become an important tool in many different fields, e.g. learning constant-depth circuits [135], disjunctive normal forms [140, 109, 122, 37], juntas [148], decision trees [128, 158], and others. It has also been previously used in communication complexity by Raz [171] and Klauck [120], and will be useful in Chapters 3 and 4. We start by defining what the Fourier expansion is. As previously mentioned, the following can be found in [157, 201].

The *Fourier expansion* of a Boolean function is a representation as a real, multilinear polynomial. The monomials in this expansion are called 'characters' or parity functions.

Definition 1.19. For $S \subseteq [n]$ we define $\chi_S : \{-1,1\}^n \to \{-1,1\}$ by

$$\chi_S(x) = \prod_{i \in S} x_i. \tag{1.5}$$

Theorem 1.20. *Every function $f : \{-1,1\}^n \to \mathbb{R}$ can be uniquely expressed as a multilinear polynomial,*

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x). \tag{1.6}$$

The real number $\widehat{f}(S)$ is called the Fourier coefficient of f on S.

Given two functions $f, g : \{-1,1\}^n \to \mathbb{R}$, an inner product $\langle \cdot, \cdot \rangle$ can be defined by

$$\langle f, g\rangle = \frac{1}{2^n} \sum_{x \in \{-1,1\}^n} f(x)g(x) = \mathop{\mathbb{E}}_{x \sim \{-1,1\}^n}[f(x)g(x)]. \tag{1.7}$$

We write $\|f\|_2 = \sqrt{\langle f, f\rangle}$, and more generally, for $p \geq 1$,

$$\|f\|_p = \left(\frac{1}{2^n} \sum_{x \in \{-1,1\}^n} |f(x)|^p\right)^{1/p}. \tag{1.8}$$

It is not hard to see that

$$\langle \chi_S, \chi_T \rangle = \begin{cases} 1 & \text{if } S = T, \\ 0 & \text{if } S \neq T, \end{cases} \tag{1.9}$$

i.e., the set of all χ_S is an orthonormal basis. Therefore we have the following formula for the Fourier coefficients.

Proposition 1.21. *For $f : \{-1,1\}^n \to \mathbb{R}$ and $S \subseteq [n]$, the Fourier coefficient $\widehat{f}(S)$ is given by*

$$\widehat{f}(S) = \langle f, \chi_S \rangle = \frac{1}{2^n} \sum_{x \in \{-1,1\}^n} f(x)\chi_S(x). \tag{1.10}$$

Another consequence of the orthonormality of the characters χ_S is *Parseval's theorem*, which states that the 2-norm of f is just the sum of the squares of f's Fourier coefficients.

Lemma 1.22 (Parseval's)**.** *For every function $f : \{-1,1\}^n \to \mathbb{R}$ we have $\|f\|_2^2 = \sum_{S \subseteq [n]} \widehat{f}(S)^2$.*

As a consequence, if f is Boolean-valued, then $\sum_{S \subseteq [n]} \widehat{f}(S)^2 = 1$.

Another important and useful concept of Boolean functions is *noise stability* [27, 156]. Informally, it is a measure of how resilient to noise a Boolean function is. Given an input $x \in \{-1,1\}^n$, one might imagine a process that flips each bit of x with some probability, which leads to some final string $y \in \{-1,1\}^n$. We then may ask about the probability that $f(x) = f(y)$, i.e., that the function's value stays intact.

Definition 1.23 ([157, Definitions 2.40 and 2.41])**.** Let $\rho \in [-1,1]$. For fixed $x \in \{-1,1\}^n$ we write $y \sim N_\rho(x)$ to denote that the random string y is drawn as follows: for each $i \in [n]$ independently,

$$y_i = \begin{cases} x_i & \text{with probability } \frac{1}{2} + \frac{1}{2}\rho, \\ -x_i & \text{with probability } \frac{1}{2} - \frac{1}{2}\rho. \end{cases} \tag{1.11}$$

We say that y is ρ-correlated to x. If $x \sim \{-1,1\}^n$ is drawn uniformly at random and then $y \sim N_\rho(x)$, we say that (x, y) is a *ρ-correlated pair* of random strings.

Given these definitions, we define the concept of noise stability, which measures the correlation between $f(x)$ and $f(y)$ when (x, y) is a ρ-correlated pair.

Definition 1.24 ([157, Definition 2.42]). For $f : \{-1,1\}^n \to \mathbb{R}$ and $\rho \in [-1,1]$, the *noise stability* of f at ρ is

$$\mathrm{Stab}_\rho[f] = \mathop{\mathbb{E}}_{\substack{(x,y)\\ \rho\text{-correlated}}} [f(x)f(y)]. \tag{1.12}$$

The noise stability of f is nicely related to f's Fourier coefficients as stated in the following theorem.

Theorem 1.25 ([157, Theorem 2.49]). *For $f : \{-1,1\}^n \to \mathbb{R}$ and $\rho \in [-1,1]$,*

$$\mathrm{Stab}_\rho[f] = \sum_{k=0}^{n} \rho^k W^k[f], \tag{1.13}$$

where $W^k[f] = \sum_{\substack{S\subseteq[n]\\ |S|=k}} \widehat{f}(S)^2$ is the Fourier weight of f at degree k.

The above result is obtained from one of the most important operators in analysis of Boolean functions: the *noise operator* T_ρ.

Definition 1.26 ([157, Definition 2.46]). For $\rho \in [-1,1]$, the *noise operator with parameter ρ* is the linear operator T_ρ on functions $f : \{-1,1\}^n \to \mathbb{R}$ defined by

$$\mathrm{T}_\rho f(x) = \mathop{\mathbb{E}}_{y\sim N_\rho(x)} [f(y)]. \tag{1.14}$$

Proposition 1.27 ([157, Proposition 2.47]). *For $f : \{-1,1\}^n \to \mathbb{R}$, the Fourier expansion of $\mathrm{T}_\rho f$ is*

$$\mathrm{T}_\rho f = \sum_{S\subseteq[n]} \rho^{|S|} \widehat{f}(S)\chi_S. \tag{1.15}$$

It is not hard to prove from the above results that $\mathrm{Stab}_\rho[f] = \langle f, \mathrm{T}_\rho f\rangle$.

We can see from Proposition 1.27 that the effect of T_ρ and, consequently, the introducing of noise, is to smear out sharp peaks in f over nearby inputs. In other words, the Fourier coefficient $\widehat{f}(S)$ is reduced by a factor $\rho^{|S|}$, which means that higher-degree Fourier coefficients are more harshly attenuated. The net effect is moving f closer to a constant function.

1.2.1 Hypercontractivity

One can show that the p-norm $\|f\|_p$ of a function is monotonic non-decreasing in p, i.e., $\|f\|_p \le \|f\|_q$ for $p \le q$. On the other hand, since $\mathrm{T}_\rho f$ is an average over functions that have the same p-norm, the triangle inequality implies that T_ρ is a contraction, i.e., $\|\mathrm{T}_\rho f\|_p \le \|f\|_p$ for every $p \ge 1$ [157, Exercise 2.33]. Bonami in 1970 [33], and Beckner later in 1975 [23], proved that this inequality still holds even if we allow the left-hand side a somewhat higher q-norm, a result now known as the *Bonami-Beckner Hypercontractive inequality* (see [157, Section 10.1]).

Theorem 1.28 (Bonami-Beckner)**.** *If $1 \leq p \leq q$ and $0 \leq \rho \leq \sqrt{(p-1)/(q-1)}$, then*

$$\|\mathrm{T}_\rho f\|_q \leq \|f\|_p. \tag{1.16}$$

The Bonami-Beckner Hypercontractive inequality is central in a lot of more advanced applications of Fourier analysis on the Boolean cube. The case $\rho = \sqrt{(p-1)/(q-1)}$ is the best possible one [157, Exercise 9.10(b)], and implies all others by monotonicity of the p-norm.

An interesting case of the above result is when either p or q equal 2, since Parseval allows us to rewrite the 2-norm in terms of Fourier coefficients [201]. If $q = 2$, $p \in [1, 2]$ and $\rho = \sqrt{p-1}$, then Theorem 1.28 implies

$$\sum_{S \subseteq [n]} (p-1)^{|S|} \widehat{f}(S)^2 = \|\mathrm{T}_\rho f\|_2^2 \leq \|f\|_p^2 = \left(\frac{1}{2^n} \sum_{x \in \{-1,1\}^n} |f(x)|^p \right)^{2/p}. \tag{1.17}$$

This inequality gives an upper bound on the Fourier coefficients such that high-degree coefficients are suppressed. An important special case is when the range of f is $\{-1, 0, 1\}$, e.g. when f is Boolean or the difference of two Boolean functions. In this case, $\|f\|_p^p = \Pr[f(x) \neq 0]$ for any p. The resulting inequality was proved by Kahn, Kalai and Linial [112] and is known as *KKL* inequality.

Lemma 1.29 (KKL)**.** *Let $f : \{-1, 1\}^n \to \{-1, 0, 1\}$ and $A = \{x | f(x) \neq 0\}$. For every $\delta \in [0, 1]$*

$$\sum_{S \subseteq [n]} \delta^{|S|} \widehat{f}(S)^2 \leq \left(\frac{|A|}{2^n} \right)^{\frac{2}{1+\delta}}. \tag{1.18}$$

On a high level, the KKL inequality tells that $\{-1, 0, 1\}$-valued functions with small support cannot have too much of their Fourier weight on small degree coefficients.

The above concepts can be generalised to matrix-valued functions, in particular the ones that map $x \in \{-1, 1\}^n$ to an m-qubit density operator. The *Fourier transform* $\widehat{f}$ of a matrix-valued function $f : \{-1, 1\}^n \to \mathbb{C}^{m \times m}$ is defined similarly as for scalar functions: is the function $\widehat{f} : \{-1, 1\}^n \to \mathbb{C}^{m \times m}$ defined by

$$\widehat{f}(S) = \frac{1}{2^n} \sum_{x \in \{-1,1\}^n} f(x) \chi_S(x). \tag{1.19}$$

Here the Fourier coefficients $\widehat{f}(S)$ are also $m \times m$ complex matrices. The equivalent of a p-norm for matrices $A \in \mathbb{C}^{m \times m}$ is the normalized Schatten p-norm

$$\|A\|_p = \left(\frac{1}{m} \operatorname{Tr} |A|^p \right)^{1/p} = \left(\frac{1}{m} \sum_{i=1}^m \sigma_i^p \right)^{1/p}, \tag{1.20}$$

where $\sigma_1, \ldots, \sigma_m$ are the singular values of A. We also define the *trace norm* $\|A\|_{\mathrm{tr}} = m\|A\|_1 = \sum_{i=1}^m \sigma_i$.

Ben-Aroya, Regev and de Wolf [24] proved in 2008 an extension of the hypercontractivity inequality to matrix-valued functions similar to Eq. (1.17), with p-norms replacing absolute values.

Theorem 1.30 ([24, Theorem 1]). *For every $f : \{-1,1\}^n \to \mathbb{C}^{m\times m}$ and $1 \le p \le 2$,*

$$\left(\sum_{S\subseteq[n]} (p-1)^{|S|}\|\widehat{f}(S)\|_p^2\right)^{1/2} \le \left(\frac{1}{2^n}\sum_{x\in\{-1,1\}^n}\|f(x)\|_p^p\right)^{1/p}. \tag{1.21}$$

A special case of the above result is the following theorem, which resembles the KKL inequality.

Theorem 1.31 ([24, Lemma 6]). *For every $f : \{-1,1\}^n \to \mathbb{C}^{2^m\times 2^m}$ and $0 \le \delta \le 1$,*

$$\sum_{S\subseteq[n]} \delta^{|S|}\|\widehat{f}(S)\|_{\mathrm{tr}}^2 \le 2^{2\delta m}. \tag{1.22}$$

The Hypercontractivity inequalities, more precisely the KKL inequality and Theorem 1.31, will be vital for our results in Chapters 3 and 4.

Chapter 2

Quantum Sketching Protocols for Hamming Distance

The concept of quantum fingerprinting, as seen in the previous chapter, was used to prove one of the first exponential classical-quantum communication separations, more specifically in the SMP model. This surprising result sparked significant interest from the perspective of computer science [209, 84] and information theory [199], as well as physics. Theoretically, it has been used to shed new light on the two-slit experiment [141] and detailed studies of fingerprinting schemes using few qubits have been undertaken [22, 174]. Proof-of-principle quantum fingerprinting experiments have been carried out with states of 1 qubit realized using linear optics [103] and nuclear magnetic resonance [64]. More recently, a variant of the quantum fingerprinting protocol based on coherent states [14] has also been implemented experimentally, surpassing the best known classical protocols [204] and even the classical theoretical limit [93].

However, equality is just one distance measure, and a very special one. In this chapter we seek other measures of distance for which quantum information can achieve a similar exponential advantage. In addition to the inherent theoretical interest of this question in terms of giving insight into the expressive power of quantum states, quantum protocols for more general distance measures could find significantly broader applications, e.g. as subroutines in cryptographic communication schemes or as means to embed other more complicate distance measures.

2.1 Our results

Our main result is a quantum protocol for approximately computing another distance measure, the Hamming distance, up to low *relative* error. This notion of accuracy is important when one wishes to compare objects that are similar; for example, when one of the objects is produced by a small number of errors affecting the other [53]. Approximating the Hamming distance

between two n-bit strings up to additive accuracy ϵn (analogous to the accuracy achieved by the protocol of [127]) would give no useful information in this situation.

In the setting we consider, Alice and Bob are given $x, y \in \{0,1\}^n$, respectively. Their goal is to approximately calculate the Hamming distance $d(x,y)$ between x and y, i.e., they must output $d_\epsilon(x,y)$ such that $(1-\epsilon)d(x,y) \leq d_\epsilon(x,y) \leq (1+\epsilon)d(x,y)$. Pang and El Gamal [163] proved a lower bound of $\Omega(n)$ for exactly calculating the Hamming distance in the multi-round two-party classical communication model. Here we describe a quantum protocol that approximately computes the Hamming distance in the SMP model by communicating $\text{poly}(\log n)$ qubits.

Theorem 2.1. *There is a quantum protocol in the SMP model with private randomness which communicates $O((\log n)^2(\log\log n)/\epsilon^3)$ qubits and computes the Hamming distance between n-bit strings up to relative error ϵ, for any $\epsilon = \Omega(1/\log n)$, with failure probability bounded above by an arbitrarily small constant.*

The protocol is based on a subroutine which determines whether, for some threshold δ, $d(x,y) \leq \delta$ or $d(x,y) \geq (1+\epsilon)\delta$. This subroutine maps x and y to N-bit strings Ax, Ay such that in the first case, $d(Ax,Ay)$ is low (less than αN, for some constant α), whereas in the second case, $d(Ax,Ay)$ is high (greater than βN, for some constant $\beta > \alpha$). Alice and Bob then encode the strings Ax and Ay as quantum fingerprints, which the referee can distinguish using the Swap test.

Note that there exists a corresponding classical protocol in the SMP model with shared randomness, with a similar complexity. One way to see this is that the quantum protocol is ultimately based on the use of the Swap test to approximately compute the inner product between unit vectors, for which there is an efficient classical protocol in this model [126].

We then generalise Theorem 2.1 to other distance measures: in particular, those which can be interpreted as distances in graphs. A graph $G = (V,E)$ is fixed in advance, and each of Alice and Bob is given a vertex of G (v and w, respectively). They aim to approximately compute $d_G(v,w)$, the length of a shortest path in G between v and w, up to relative error ϵ.

We first observe that Theorem 2.1 can be applied to give an efficient protocol for this problem whenever there is a distance-preserving embedding of G into the hypercube: the graph whose vertex set is $\{-1,1\}^m$, for some m, and where two vertices are connected by an edge whenever their Hamming distance is 1. In fact, this can be generalised further, to graphs which are embeddable into the hypercube such that distances are preserved up to a constant factor k. Such graphs are known as ℓ_1-graphs, because it turns out that this criterion is equivalent to the existence of a distance-preserving embedding of the graph in ℓ_1 [18]. The class of ℓ_1-graphs includes all trees, median graphs, Hamming graphs, and Johnson graphs [18]. (We include in the Appendix 2.A a characterization of ℓ_1-graphs which we were not able to find in the literature.)

Distances in ℓ_1-graphs are used in a variety of applications, a few of which we outline here. Partial cubes (ℓ_1-graphs with embedding constant $k = 1$) were initially introduced by Graham and Pollak [89] as a model for interconnection networks in the Bell System, with

distances between vertices corresponding to the number of hops between 'loops' in their network. Antimatroids (a specific subclass of ℓ_1-graphs) are used as structures to represent the required steps to develop a student's knowledge in a certain topic, and the distance between two points that represent concepts in these structures corresponds to the length of a student's learning path [71]. The Barnes-Hut tree method in many-body physics [21] provides a systematic way of determining the degree of 'closeness' between two different particles. The distance between two nodes in the tree is linked to this 'closeness' property and can be used for various purposes, e.g. to calculate gravitational forces in star clusters and study galaxy evolution [167]. Tree structures are also used in biology, where phylogenetic trees classify organisms based on overall similarity, and the distance between vertices is related to genetic or mutation distance [77].

Our protocol is efficient for ℓ_1-graphs G whose diameter $\mathrm{diam}(G)$ is low, where the diameter is defined as $\mathrm{diam}(G) = \max_{v,w} d_G(v,w)$.

Theorem 2.2. *Let $G = (V, E)$ be an ℓ_1-graph with $|V|$ vertices, and let $v, w \in V$. There is a quantum protocol in the SMP model with private randomness which computes $d_G(v,w)$ up to relative error ϵ, for any $\epsilon = \Omega(1/\log \mathrm{diam}(G))$, with failure probability bounded above by an arbitrarily small constant and communicates $O((\log \mathrm{diam}(G))(\log\log \mathrm{diam}(G))(\log\log |V|)/\epsilon^3)$ qubits*

For any graph G, even testing equality between vertices requires $\Omega(\sqrt{\log |V|})$ bits of classical communication in the SMP model without shared randomness [8], so this is an exponential separation for those ℓ_1-graphs where $\mathrm{diam}(G) = O(\log |V|)$, e.g. expander graphs. $d_G(v,w)$ can be computed trivially using $O(\log |V|)$ bits of classical communication, by sending the labels of v and w to the referee. So for graphs G where $\mathrm{diam}(G)$ is close to $|V|$, Theorem 2.2 gives little or no improvement on the classical complexity. One may wonder whether this is simply a limitation of our protocol, but we show that this is not the case.

Theorem 2.3. *Given a graph G with diameter $\mathrm{diam}(G)$, any one-way quantum communication protocol that computes $d_G(v,w)$ up to relative error $\epsilon < 1/4$ with failure probability $1/3$ must transmit at least $\Omega(\log \mathrm{diam}(G))$ qubits.*

As every protocol in the SMP model implies a one-way protocol, this shows that the complexity of our protocol is nearly optimal in terms of its dependence on $\mathrm{diam}(G)$.

Finally, we show that our protocol for approximately computing the Hamming distance can be used to give an efficient protocol for approximately computing the ℓ_1-distance between vectors in $\mathbb{R}^n$.

Theorem 2.4. *Let $x, y \in [-1,1]^n$ such that each entry of x and y is specified by a k-bit string, with $k = O(\log n)$. There is a quantum protocol in the SMP model with private randomness which communicates $O((\log n)^2(\log\log n)/\epsilon^3)$ qubits and computes $\|x - y\|_1$ up to relative error ϵ, for any $\epsilon = \Omega(1/\log n)$, with failure probability bounded above by an arbitrarily small constant.*

A natural special case of Theorem 2.4 is where x and y are probability distributions. Then our result enables Alice and Bob to determine the distance between two distributions, one of which is a small perturbation of the other.

2.1.1 Related work

The Hamming distance is a fundamental distance measure and has been studied in various forms. In the context of quantum communication complexity, Liu and Zhang [136] gave a quantum sketching protocol for the related "threshold" problem of determining whether the Hamming distance is larger than d, for some d. Their protocol uses $O(d \log n)$ communication, improving a previous $O(d \log^2 n)$ protocol of Gavinsky, Kempe and de Wolf [83]. Huang *et al.* [104] had previously proven an $\Omega(d)$ lower bound for even the two-way quantum communication complexity of the threshold Hamming distance problem, together with an $O(d \log d)$ upper bound in the classical SMP model with public randomness.

A key ingredient in the upper bound of Huang *et al.* is a protocol which communicates $O(1)$ bits and distinguishes between the case that the Hamming distance is at most d, and the case that the Hamming distance is at least $2d$, for arbitrary d. Their protocol can be seen as a variant of our Lemma 2.6 below with $N = 1$; similar analysis shows that it could be generalised to distinguish between Hamming distance d and Hamming distance $(1+\epsilon)d$ with $O(1/\epsilon^2)$ bits of communication. Using a generic construction of Yao [209], improved by Gavinsky, Kempe, and de Wolf [84], this implies a quantum sketching protocol for the same task which communicates $2^{O(1/\epsilon^2)} \log n$ qubits. Using a similar approach to our work, this in turn implies a protocol which solves the approximate Hamming distance problem by transmitting $2^{O(1/\epsilon^2)} \operatorname{poly} \log n$ qubits. This is the same asymptotic complexity as our protocol for constant ϵ, but in practice the $2^{O(1/\epsilon^2)}$ factor makes the protocol infeasible for even modest values of ϵ.

Classically, there has also been substantial work on approximately computing the Hamming distance between a small "pattern" and a larger string, both locally and in a distributed context (see [49] and references therein).

2.2 The Protocol

In this section we present our protocol for approximating the Hamming distance $d(x, y)$ between two strings $x, y \in \{0,1\}^n$ up to relative error ϵ in the SMP model. That is, Alice and Bob seek the referee to output $d_\epsilon(x, y)$ such that $(1-\epsilon)d(x, y) \leq d_\epsilon(x, y) \leq (1+\epsilon)d(x, y)$. Call this problem HAM_ϵ.

We first state a lemma that is going to be useful for our protocol and which encapsulates results on quantum fingerprinting by Yao [209]. Recall the quantum fingerprint states $|h_x\rangle$ from Definition 1.17.

Lemma 2.5. *Given $x, y \in \{0,1\}^N$, their Hamming distance $d(x,y)$ can be estimated up to additive accuracy $N\epsilon$ with failure probability δ using $O(\log(1/\delta)/\epsilon^2)$ copies of $|h_x\rangle$ and $|h_y\rangle$.*

Proof. First note that

$$\langle h_y|h_x\rangle = \frac{1}{N}\sum_{i=1}^{N}\langle y_i|x_i\rangle = 1 - \frac{d(x,y)}{N}. \tag{2.1}$$

Recall that the Swap test outputs either 0 or 1 on input $|h_x\rangle|h_y\rangle$, and outputs 1 with probability

$$p := \frac{1}{2}\left(1 - |\langle h_y|h_x\rangle|^2\right) = \frac{d(x,y)}{N} - \frac{d(x,y)^2}{2N^2}. \tag{2.2}$$

Before presenting our protocol, we first analyse two different cases: $d(x,y)/N \leq 9/10$ and $d(x,y)/N > 9/10$.

Suppose that $d(x,y)/N$ is bounded away from 1, say $d(x,y)/N \leq 9/10$. This means that p is bounded away from $1/2$. We apply the swap test to k copies of $|h_x\rangle|h_y\rangle$, for some k to be determined. Let P_i correspond to the outcome of the i-th swap test and $P := \frac{1}{k}\sum_{i=1}^{k} P_i$. By a Chernoff bound [65, Theorem 1.1],

$$\Pr[|P - p| \geq \epsilon] \leq 2e^{-2k\epsilon^2}, \tag{2.3}$$

which means that we obtain an approximation $P = p \pm \epsilon$ with probability $1-\delta$ by taking $k = O(\log(1/\delta)/\epsilon^2)$. Since $d(x,y)/N = 1 - \sqrt{1-2p}$, we set $\widetilde{d}/N = 1 - \sqrt{1-2P}$ as our approximation to $d(x,y)/N$. The derivative of $\sqrt{1-2z}$ is $1/\sqrt{1-2z}$, which is $O(1)$ around $z = p$, since p is bounded away from $1/2$. Then, by a Taylor expansion around p, we have that, with probability $1-\delta$,

$$\left|\frac{\widetilde{d}}{N} - \frac{d(x,y)}{N}\right| = O(|P-p|) = O(\epsilon). \tag{2.4}$$

On the other hand, if $d(x,y)/N$ is close to 1, say $d(x,y)/N > 9/10$, then we use the following result due to Yao [209, Lemma 1] to bound the outcome of the swap tests,

$$\Pr\left[\left|\tilde{d} - d(x,y)\right| \geq N\epsilon\right] \leq 2e^{-k\epsilon^4/32}, \tag{2.5}$$

where $\tilde{d}/N = 1 - \sqrt{1-2P}$, with again $P := \frac{1}{k}\sum_{i=1}^{k} P_i$. Hence, if $k = O(\log(1/\delta)/\epsilon^2)$, with probability $1-\delta$ we obtain an approximation $\widetilde{d}/N = d/N \pm \sqrt{\epsilon}$.

We now present our protocol for estimating $d(x,y)/N$: fix $k = O(\log(1/\delta)/\epsilon^2)$. For half of the k copies we apply the usual swap test. If the resulting estimate $\widetilde{d}/N$ is at most $2/3$, then we output it as the final answer. Otherwise, by using the other half of the k copies, we apply the swap tests on $\mathbb{I}\otimes X|h_x\rangle$ and $|h_y\rangle$, where X is the usual Pauli operator, and obtain an estimate $\widetilde{d}/N$ (the referee can apply the operator $\mathbb{I}\otimes X$ to his copies of $|h_x\rangle$). We output $1 - \widetilde{d}/N$ as the final answer.

To see why this works, first note that, if $d(x,y)/N > 9/10$, then its approximation from the swap tests is such that $\widetilde{d}/N > 9/10 - \sqrt{\epsilon} > 2/3$ for sufficiently small ϵ and probability $1-\delta/2$. This means that, if the estimate from the first half of the copies is $\leq 2/3$, then it must be an approximation to some $d(x,y)/N \leq 9/10$, and we know that it is within distance ϵ with probability $1-\delta/2$.

On the other hand, if the estimate of the first half of the copies is $> 2/3$, then $d(x,y)/N \geq 2/3 - \epsilon > 1/2$ for sufficiently small ϵ and with probability $1-\delta/2$. Note that $\langle h_y|\mathbb{I}\otimes X|h_x\rangle = d(x,y)/N$, i.e., estimating the Hamming distance d/N with $\mathbb{I}\otimes X|h_x\rangle$ and $|h_y\rangle$ is equivalent to estimating the Hamming distance $1-d/N$ with $|h_x\rangle$ and $|h_y\rangle$. This means that the swap tests on the second half of the copies will estimate the Hamming distance $1-d(x,y)/N \leq 1/2$, for which an approximation $\widetilde{d}/N$ within distance ϵ can be obtained with probability $1-\delta/2$. Thus, with overall probability $1-\delta$, our protocol outputs $\widetilde{d}$ such that $|\widetilde{d}-d| \leq \epsilon N$ in both cases. ■

Remark. Given that we aim to approximately compute the inner product between $|h_x\rangle$ and $|h_y\rangle$ in Lemma 2.5, the reader may wonder why the Hadamard test [6] was not used instead, given that this test allows direct estimation of $\langle h_y|h_x\rangle$. The reason is that the Hadamard test requires the ability to produce the coherent superposition $\frac{1}{\sqrt{2}}(|0\rangle|h_x\rangle + |1\rangle|h_y\rangle)$, which is not available to the referee.

In the following, we use the notation $|z|$ to mean the number of entries equal to 1 in a string $z \in \{0,1\}^n$.

Lemma 2.6. *Consider an $N \times n$ matrix A over $\mathbb{F}_2$ whose entries are randomly chosen from $\{0,1\}$, and equal to 1 with independent probability $1/(2d)$ for some $d \geq 1$. Fix $\epsilon > 0$. Then there exist values $\delta_1 < \delta_2$ that do not depend on N and n, such that $\delta_2 - \delta_1 = \Theta(\epsilon)$ and for any $\eta > 0$:*

- *for all $z \in \{0,1\}^n$ such that $|z| \leq d$, $\Pr_A\left[|Az| \geq N\delta_1 + N\eta\right] \leq e^{-2N\eta^2}$;*
- *for all $z \in \{0,1\}^n$ such that $|z| \geq (1+\epsilon)d$, $\Pr_A\left[|Az| \leq N\delta_2 - N\eta\right] \leq e^{-2N\eta^2}$.*

Hence, for sufficiently large $N = \Theta(n/\epsilon^2)$, with high probability over the choice of A, it is sufficient to determine $|Az|$ up to additive accuracy $\Theta(N\epsilon)$ to distinguish between the cases $|z| \leq d$ and $|z| \geq (1+\epsilon)d$.

Proof. It is shown in [130, Lemma 2.1] that for any $z \in \{0,1\}^n$ and $i \in [N]$, $\Pr[(Az)_i = 1] = \frac{1}{2}\left(1-(1-1/d)^{|z|}\right)$ and that the probabilities of this event for $|z| \leq d$ and $|z| \geq (1+\epsilon)d$ are bounded by values δ_1, δ_2 that do not depend on N and n and are separated by $\Theta(1-e^{-\epsilon}) = \Theta(\epsilon)$. That is,

$$\Pr_A\left[(Az)_i = 1\right] \leq \delta_1 = \frac{1}{2}\left(1 - \left(1-\frac{1}{d}\right)^d\right) \qquad \text{if } |z| \leq d, \tag{2.6a}$$

$$\Pr_A\left[(Az)_i = 1\right] \geq \delta_2 = \frac{1}{2}\left(1 - \left(1-\frac{1}{d}\right)^{(1+\epsilon)d}\right) \quad \text{if } |z| \geq (1+\epsilon)d. \tag{2.6b}$$

The expected value of $|Az| = \sum_i (Az)_i$ then satisfies

$$\mathbb{E}[|Az|] \leq N\delta_1 \quad \text{if } |z| \leq d, \tag{2.7a}$$

$$\mathbb{E}[|Az|] \geq N\delta_2 \quad \text{if } |z| \geq (1+\epsilon)d. \tag{2.7b}$$

If $|z| \leq d$ so that $\mathbb{E}[|Az|] \leq N\delta_1$, by a Chernoff bound (e.g. [65, Theorem 1.1]) we obtain

$$\Pr_A\big[|Az| \geq N\delta_1 + N\eta\big] \leq e^{-2N\eta^2}. \tag{2.8}$$

By the same token, if $|z| \geq (1+\epsilon)d$, so that $\mathbb{E}|Az|] \geq N\delta_2$, we obtain

$$\Pr_A\big[|Az| \leq N\delta_2 - N\eta\big] \leq e^{-2N\eta^2}. \tag{2.9}$$

Taking a union bound over all $z \in \{0,1\}^n$ in both cases, we have

$$\left.\begin{aligned} \Pr_A\big[\exists z \text{ s.t. } |z| \leq d \text{ and } |Az| \geq N\delta_1 + N\eta\big] \\ \Pr_A\big[\exists z \text{ s.t. } |z| \geq (1+\epsilon)d \text{ and } |Az| \leq N\delta_2 - N\eta\big] \end{aligned}\right\} \leq 2^n e^{-2N\eta^2} = e^{n\ln 2 - 2N\eta^2}, \tag{2.10}$$

so that it is sufficient to choose $N = \Omega(n/\eta^2)$ to bound the probability that either case occurs by an arbitrarily small constant. Choosing $\eta = c\epsilon$ for a sufficiently small constant c, we have $|Az| \leq N(\delta_1 + c\epsilon)$ if $|z| \leq d$, and $|Az| \geq N(\delta_2 - c\epsilon)$ if $|z| \geq (1+\epsilon)d$. Therefore, it is sufficient to determine $|Az|$ up to additive accuracy $O(N\epsilon)$ to distinguish these two cases. ■

The map A in Lemma 2.6 can be interpreted as a linear code. Here we choose the matrix A to be sparse and random, which enables us to control its behaviour when acting on strings z such that $|z| \approx d$ for small d.

We now describe our protocol based on the two previous Lemmas. In this protocol, Alice and Bob have already agreed beforehand on the matrix A, guaranteed to exist by Lemma 2.6, to be used. We stress that this matrix is fixed in advance and does not need to be chosen using shared randomness.

Protocol 2.7. *Consider the following subroutine for arbitrary $d \in [1,n]$ and $\delta > 0$: Alice and Bob encode their n-bit strings x and y as Ax and Ay, respectively, where A is picked according to Lemma 2.6 and multiplication is over $\mathbb{F}_2$. They send $O((\log 1/\delta)/\epsilon^2)$ copies of the quantum states $|h_{Ax}\rangle$ and $|h_{Ay}\rangle$ to the referee, who performs Swap tests and estimates the Hamming distance $d(Ax, Ay)$ up to accuracy $N\epsilon$ with failure probability δ. By Lemma 2.6, this is sufficient to determine whether $d(x,y) \leq d$ or $d(x,y) \geq (1+\epsilon)d$ with failure probability δ.*

Alice and Bob then apply this subroutine to the sequence S of values d

$$0, 1, 1+\epsilon, (1+\epsilon)^2, \ldots \tag{2.11}$$

where the last element in S corresponds to the minimal k such that $(1+\epsilon)^{k+1} > n$; there are $O(\log n/\log(1+\epsilon)) = O((\log n)/\epsilon)$ elements in the sequence. (In the case $d = 0$, they use the standard quantum fingerprinting protocol instead.) Given the $O((\log n)/\epsilon)$ results, the referee outputs the minimal $\widetilde{d}$ such that the subroutine returned "$d(x,y) \leq \widetilde{d}$".

We first show that, if each use of the subroutine succeeds, the overall algorithm achieves the required level of accuracy. By the definition of S, there exist consecutive elements $d_0, d_1, d_2 \in S$ such that $d_0 \leq d(x,y)/(1+\epsilon)$, $d(x,y)/(1+\epsilon) \leq d_1 \leq d(x,y)$, $d(x,y) \leq d_2 \leq (1+\epsilon)d(x,y)$. Then on input d_2 the subroutine must return "$d(x,y) \leq d_2$", while for input d_0 it must return "$d(x,y) \geq (1+\epsilon)d_0$", so the output $\widetilde{d}$ is either d_1 or d_2 and hence

$$(1-\epsilon)d(x,y) \leq \frac{d(x,y)}{1+\epsilon} \leq \widetilde{d} \leq (1+\epsilon)d(x,y). \tag{2.12}$$

Setting $\delta = O(\epsilon/\log n)$ and using a union bound over the $O((\log n)/\epsilon)$ uses of the subroutine, the probability that any of the subroutines fails can be upper-bounded by an arbitrarily small positive constant.

The overall communication complexity is

$$O\left(\underbrace{((\log n)/\epsilon)}_{\#\ d\text{'s}} \cdot \underbrace{(\log(1/\delta)/\epsilon^2)}_{\#\ \text{Swap tests}} \cdot \underbrace{(\log(n/\epsilon))}_{\#\ \text{qubits}\ |h_{Ax}\rangle}\right) = O\left((\log n)^2(\log\log n)/\epsilon^3\right), \tag{2.13}$$

assuming that $\epsilon \geq 1/\log n$. This completes the proof of Theorem 2.1.

2.3 Measuring Distances in Graphs

In the following, for an arbitrary graph G and vertices v, w, let $d_G(v,w)$ denote the distance between v and w in G, i.e., the length of a shortest path between v and w. Also, the hypercube graph Q_n is defined as the graph with vertex set $\{0,1\}^n$, where distance between vertices is the Hamming distance.

The algorithm from last section for approximately measuring the Hamming distance between two strings in the SMP model can be slightly modified to approximately compute the distance between two vertices in specific graphs in the SMP model. That is, to solve the following problem: for some graph $G = (V,E)$, and given vertices v, w as input, output $\widetilde{d}$ such that $(1-\epsilon)d_G(v,w) \leq \widetilde{d} \leq (1+\epsilon)d_G(v,w)$. Call this problem $\mathrm{DIS}_\epsilon[G]$. The idea is to embed a given graph G into a hypercube graph such that all the distances between vertices are preserved or rescaled by a constant factor. Once this embedding is achieved, the hypercube structure allows the equivalence between vertex distance in the graph and Hamming distance, so that a binary string can be associated with each vertex and the algorithm can be applied to these binary strings.

The downside of the above approach is that it cannot be applied to any given graph, since most graphs are not isometrically embeddable into a hypercube. The graphs which can be isometrically embedded into hypercubes are known as partial cubes [34, 161].

The identification of which graphs are partial cubes is an interesting question by itself. The class of partial cubes is relatively broad. The most important examples are hypercubes, trees [203] and median graphs [162]. It also includes other significant classes, e.g. tope graphs of oriented

matroids (specially graphs of regions of hyperplane arrangements) [31, 70], antimatroids [113, 70], weak orderings [70], bipartite $(6,3)$-graphs [18], tiled partial cubes [35] and netlike partial cubes [168].

Partial cubes can be fully characterized via *Djoković's Characterization* [62, 60], introduced by Djoković in 1973. It connects the property of isometric embedding to bipartiteness and convexity of some specific sub-graphs of the original graph. Here a set is said to be convex if it is closed under taking shortest paths, i.e., if the shortest paths between any two points from the set are also contained in the set. Djoković's Characterization states, more specifically, that a connected graph G can be isometrically embedded into a hypercube if and only if G is bipartite and $G(a|b)$ is convex for each edge (a,b) of G, where $G(a|b) := \{x \in V(G) |\ d_G(x,a) < d_G(x,b)\}$ is the set of the vertices closer to a than b. In other words, to check if a graph is a partial cube, one needs to check first if the graph is bipartite. Apart from that, one chooses an edge and constructs the set of all vertices that are closer to one vertex of the chosen edge than the other vertex. Then one needs to check if all shortest paths connecting any two vertices from this set only pass through the vertices of the set. If yes, the set is said to be convex and the same procedure is repeated for another edge of the original graph. If all sets constructed in this way are convex, then the graph is a partial cube.

Since the original protocol is unaffected if all distances are rescaled by a constant factor, the idea of partial cubes can be expanded by the following definitions.

Definition 2.8 ([18, 181])**.** Given two connected and unweighted graphs G and H, we write $G \overset{k}{\hookrightarrow} H$ and say that G is a k-scale embedding of H if there exists a mapping $\sigma : V(G) \to V(H)$ such that $d_H(\sigma(a), \sigma(b)) = k \cdot d_G(a,b)$ for all nodes $a, b \in V(G)$.

It is clear that partial cubes are just graphs which can be embedded in a hypercube with a 1-scale embedding. An example of a graph which is not a partial cube, but can be embedded in a hypercube with a k-scale embedding for $k > 1$, is a triangle, which embeds into Q_3 with $k = 2$.

Definition 2.9 ([60, Proposition 4.2.2])**.** A graph G is said to be an ℓ_1-graph if its path metric d_G is ℓ_1-embeddable, i.e., there is a map $f : V(G) \to \mathbb{R}^m$, for some m, such that $d_G(v,w) = \|f(v) - f(w)\|_1$.

Theorem 2.10 ([32] and [18, Theorem 8.3])**.** *A graph G is an ℓ_1-graph iff it admits a scale embedding into a hypercube.*

This means that the graphs we are interested in are ℓ_1-graphs. This class of ℓ_1-graphs includes new graphs that are not partial cubes, e.g. Hamming graphs, half cubes and Johnson graphs are 2-embeddable into a hypercube [18]. In the Appendix 2.A we developed a similar characterization for ℓ_1-graphs and the final result is the following theorem, which is Djoković's characterization without the bipartite requirement.

Theorem 2.11. *A graph G is an ℓ_1-graph iff $G(a|b)$ is convex for each edge (a,b) of G.*

By allowing the rescaling of all the distances by an even factor we can relax the bipartite requirement, but not the convexity of the $G(a|b)$ subgraphs. As an example of a direct consequence of the above result, it is known that graphs of the form C_{2n} and $C_{2n}\square K_2$ for $n \geq 2$ are partial cubes [34], where C_n is a cycle on n vertices, K_n is the complete graph with n vertices, and $\square$ denotes the Cartesian product; therefore all graphs of the form C_n and $C_n\square K_2$, for $n \geq 2$, are ℓ_1-graphs.

Before stating the communication protocol in the SMP model to approximately measure the distance between two vertices in an ℓ_1-graph, we state the Johnson-Lindenstrauss lemma [110, 56, 84], which is going to be useful to reduce the protocol complexity. Note that we use Dirac notation for vectors which are not necessarily normalized.

Lemma 2.12 (Johnson-Lindenstrauss lemma)**.** *Consider $0 < \epsilon < 1/2$ and a positive integer n. Then for any set U of k vectors in $\mathbb{R}^n$, there is a linear map $f : \mathbb{R}^n \to \mathbb{R}^{O((\log k)/\epsilon^2)}$ such that for all $|u\rangle, |v\rangle \in U$,*

$$(1-\epsilon)\||u\rangle - |v\rangle\|^2 \leq \|f|u\rangle - f|v\rangle\|^2 \leq (1+\epsilon)\||u\rangle - |v\rangle\|^2. \tag{2.14}$$

To find a map f achieving the bounds of Lemma 2.12, one can choose it at random from an appropriate distribution. A number of different constructions of such random maps are known; one simple example is a suitably normalised projection onto a random subspace of $\mathbb{R}^n$.

As mentioned, e.g. in [84], if the set U includes the 0-vector, then the map f also approximately preserves the inner product between all the pairs of vectors in U up to additive error. This implies the following lemma.

Lemma 2.13. *Let $0 < \epsilon < 1/2$. Let U be a set of unit vectors in $\mathbb{R}^n$ and let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a linear map such that, for all $|u\rangle, |v\rangle \in U \cup \{\vec{0}\}$,*

$$(1-\epsilon)\||u\rangle - |v\rangle\|^2 \leq \|f|u\rangle - f|v\rangle\|^2 \leq (1+\epsilon)\||u\rangle - |v\rangle\|^2. \tag{2.15}$$

Define the unit vectors $|\widetilde{u}\rangle = f|u\rangle/\|f|u\rangle\|$ for all $|u\rangle \in U$. Then

$$\Big|\big|\langle\widetilde{u}|\widetilde{v}\rangle\big| - \big|\langle u|v\rangle\big|\Big| \leq 4\epsilon \tag{2.16}$$

for all $|u\rangle, |v\rangle \in U$.

Proof. For clear notation, define $|u'\rangle := f|u\rangle$. By the conditions on f, we have that

$$\begin{cases} 1-\epsilon \leq \langle u'|u'\rangle \leq 1+\epsilon, \\ (1-\epsilon)\||u\rangle - |v\rangle\|^2 \leq \||u'\rangle - |v'\rangle\|^2 \leq (1+\epsilon)\||u\rangle - |v\rangle\|^2 \end{cases} \tag{2.17}$$

for all $|u\rangle, |v\rangle \in U$, where the first line was obtained by taking the 0-vector as one of the vectors and using linearity of f. From the above inequalities it follows that

$$(1+\epsilon)\langle u|v\rangle - 2\epsilon \leq \langle u'|v'\rangle \leq (1-\epsilon)\langle u|v\rangle + 2\epsilon. \tag{2.18}$$

These new inequalities in turn lead to

$$\langle\widetilde{u}|\widetilde{v}\rangle \geq \frac{(1+\epsilon)\langle u|v\rangle - 2\epsilon}{1+\epsilon} \geq \langle u|v\rangle - 2\epsilon, \tag{2.19a}$$

$$\langle\widetilde{u}|\widetilde{v}\rangle \leq \frac{(1-\epsilon)\langle u|v\rangle + 2\epsilon}{1-\epsilon} \leq \langle u|v\rangle + 4\epsilon, \tag{2.19b}$$

using that $0 < \epsilon < 1/2$. Therefore

$$\Big|\big|\langle\widetilde{u}|\widetilde{v}\rangle\big| - \big|\langle u|v\rangle\big|\Big| \leq \Big|\langle\widetilde{u}|\widetilde{v}\rangle - \langle u|v\rangle\Big| \leq 4\epsilon. \tag{2.20}$$

■

Consider applying Lemma 2.5 to the normalized quantum states $|\widetilde{h}_x\rangle$ and $|\widetilde{h}_y\rangle$ that are produced by using the Johnson-Lindenstrauss lemma, in the sense that the original states $|h_x\rangle$, $|h_y\rangle$ in Lemma 2.5 are replaced with the states $|\widetilde{h}_x\rangle$, $|\widetilde{h}_y\rangle$. We argue that this does not change the parameters of the lemma substantially. To see that, we note $\big|\widetilde{\eta} - |\langle h_y|h_x\rangle|\big| + \big||\langle\widetilde{h}_y|\widetilde{h}_x\rangle| - |\langle h_y|h_x\rangle|\big| \geq \big|\widetilde{\eta} - |\langle\widetilde{h}_y|\widetilde{h}_x\rangle|\big|$ and hence $\big|\widetilde{\eta} - |\langle\widetilde{h}_y|\widetilde{h}_x\rangle|\big| \geq 5\epsilon \implies \big|\widetilde{\eta} - |\langle h_y|h_x\rangle|\big| \geq \epsilon$, which means

$$\Pr\big[\big|\widetilde{\eta} - |\langle\widetilde{h}_y|\widetilde{h}_x\rangle|\big| \geq 5\epsilon\big] \leq \Pr\big[\big|\widetilde{\eta} - |\langle h_y|h_x\rangle|\big| \geq \epsilon\big], \tag{2.21}$$

where $\widetilde{\eta}$ is as defined in Lemma 2.5.

With this in mind, and recalling that $\mathrm{diam}(G)$ is defined to be the diameter of the graph G, i.e., the greatest distance between any pair of vertices, we present the communication protocol.

Protocol 2.14. *Alice and Bob each hold vertices $u, v \in V(G)$, respectively, from a graph G which admits a k-scale embedding into a hypercube Q_n, for some n. Their vertex images are the n-bit strings $x, y \in Q_n$, respectively. The communication protocol to measure $(1 \pm \epsilon)d_G(u, v)$ can be divided into three parts.*

First, given $d \in [1, \mathrm{diam}(G)]$ and a matrix A picked according to Lemma 2.6, Alice and Bob encode their n-bit strings x and y as Ax and Ay, respectively, where multiplication is over $\mathbb{F}_2$. Differently from the original protocol, Alice and Bob apply the Johnson-Lindenstrauss lemma to their data Ax and Ay, which are then encoded into the quantum states $|\widetilde{h}_{Ax}\rangle$ and $|\widetilde{h}_{Ay}\rangle$. There are $|V|$ possible vectors to encode, so the number of qubits to be used is reduced from $O(\log n + \log(1/\epsilon))$ to $O(\log\log|V| + \log(1/\epsilon))$.

Second, Alice and Bob send $O((\log 1/\delta)/\epsilon^2)$ copies of their quantum states $|\widetilde{h}_{Ax}\rangle$ and $|\widetilde{h}_{Ay}\rangle$ to the referee, who performs Swap tests and estimates the Hamming distance $d(Ax, Ay)$ up to accuracy $N\epsilon$ with failure probability δ, and from this decides if $d(x, y) \leq d$ or $d(x, y) \geq (1+\epsilon)d$.

The third and final part is to apply the first and second parts to the sequence S of values d

$$0, 1, 1+\epsilon, (1+\epsilon)^2, \ldots \tag{2.22}$$

where the last element in S corresponds to the minimal k such that $(1+\epsilon)^{k+1} > \mathrm{diam}(G)$; there are $O((\log\mathrm{diam}(G))/\epsilon)$ elements in the sequence. Based on the results from the Swap tests, the referee outputs $\widetilde{d}$ such that $(1-\epsilon)d(x, y) \leq \widetilde{d} \leq (1+\epsilon)d(x, y)$, in the same way as in Protocol 2.7.

Setting $\delta = O(\epsilon/\log \mathrm{diam}(G))$, the overall communication complexity is then

$$O((\log \mathrm{diam}(G))(\log\log \mathrm{diam}(G))(\log\log |V|)/\epsilon^3), \tag{2.23}$$

assuming that $\epsilon \geq 1/(\log \mathrm{diam}(G))$.

The performance of the protocol is limited by the diameter of the graph. It is known that $\log_{\Delta-1} |V| - \frac{2}{\Delta} \leq \mathrm{diam}(G) < |V|$, where Δ is the maximum vertex degree [48]. If $\mathrm{diam}(G) = O(\log |V|)$, the overall complexity is polyloglog in $|V|$. On the other hand, if $\mathrm{diam}(G) = \Theta(|V|)$, the overall complexity is polylog in $|V|$, which is no better than the trivial protocol where Alice and Bob send their entire inputs to the referee.

2.3.1 Lower bound

One can ask if there could exist other protocols substantially more efficient than ours. In order to answer this, we prove a lower bound on the quantum communication complexity for the problem of approximately calculating the graph distance between two vertices on a graph, which demonstrates that our protocol is essentially optimal in terms of the dependence of its complexity on the graph diameter. We do not know whether the 3th-power dependence on ϵ is optimal, and suspect that it may not be.

The idea behind our proof is to transform the approximate graph distance problem into the problem of approximating the modulus of the difference between two integers. We then show that two uses of a protocol for this approximate modulus problem can compute the Greater-Than function in the one-way communication model. It was shown by Zhang [211] that the one-way quantum communication complexity of this problem is maximal, improving a previous lower bound of Klauck [119] by a logarithmic term. The bound of [211] is used to obtain the lower bound for the approximate modulus problem, and hence for the approximate graph distance problem.

The first step of our proof is to show that two uses of a protocol for the approximate modulus problem can solve the Greater-Than function in the one-way communication model. Consider the Greater-Than problem (GT) defined by the Boolean function $\mathrm{GT} : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ as

$$\mathrm{GT}(x,y) = \begin{cases} 1 \text{ if } x \geq y, \\ 0 \text{ if } x < y, \end{cases} \tag{2.24}$$

where x and y are interpreted as m-bit integers. Given $0 \leq \epsilon < 1$, consider the approximate modulus problem where Alice and Bob are each given an integer x and y (respectively), each expressed as an m-bit string, and seek to output $\widetilde{d}$ such that $(1-\epsilon)|x-y| \leq \widetilde{d} \leq (1+\epsilon)|x-y|$. Call this problem MOD_ϵ. In the following we prove that two uses of this protocol suffice to solve the GT problem.

Lemma 2.15. *For any $\epsilon < 1/4$, $Q^1(GT) = O(Q^1(MOD_\epsilon))$.*

Proof. Let $\mathcal{P}_{MOD}$ be a communication protocol for MOD_ϵ in the one-way communication model with failure probability 1/6. (We can obtain a protocol which achieves this failure probability and communicates $O(Q^1(\mathrm{MOD}_\epsilon))$ qubits using $O(1)$ repetitions of the protocol which achieves failure probability 1/3 and communicates $Q^1(\mathrm{MOD}_\epsilon)$ qubits.)

Two uses of $\mathcal{P}_{MOD}$ suffice to obtain a communication protocol for GT in the one-way communication model with failure probability 1/3 as follows: Alice and Bob apply the protocol $\mathcal{P}_{MOD}$ using x and y as inputs and Bob obtains $z_0 \in [(1-\epsilon)|x-y|, (1+\epsilon)|x-y|]$. They both apply the same protocol again, but now Bob inputs $y+z_0$ (Alice still inputs x). Bob obtains z_1. If $z_0 < z_1$, then $x < y$ and he outputs 0. Otherwise, $x \geq y$ and he outputs 1.

To see why this protocol works (assuming that each use of $\mathcal{P}_{MOD}$ succeeds), note that if $x < y$, then $(2-\epsilon)|x-y| \leq |x-y-z_0| \leq (2+\epsilon)|x-y|$, and hence

$$(2-\epsilon)(1-\epsilon)|x-y| \leq z_1 \leq (2+\epsilon)(1+\epsilon)|x-y|. \tag{2.25}$$

If $x \geq y$, then $0 \leq |x-y-z_0| \leq \epsilon|x-y|$, and hence

$$0 \leq z_1 \leq \epsilon(1+\epsilon)|x-y|. \tag{2.26}$$

For $x < y$ we want to have $z_0 < z_1$, i.e., $1+\epsilon < (2-\epsilon)(1-\epsilon)$, which holds if $\epsilon < 2-\sqrt{3}$. And for $x \geq y$ we need $z_0 \geq z_1$, i.e., $\epsilon(1+\epsilon) \leq 1-\epsilon$, which holds if $\epsilon \leq \sqrt{2}-1$. Therefore, by taking $\epsilon < 1/4$, for example, one can distinguish the cases $x < y$ and $x \geq y$ through a comparison between z_0 and z_1.

Given that every protocol for MOD_ϵ in the one-way communication model implies a protocol for GT, we conclude that $Q^1(\mathrm{GT}) = O(Q^1(\mathrm{MOD}_\epsilon))$. ■

The next step is to reduce the approximate graph distance problem to the approximate modulus problem, which we achieve as follows. Let G be a graph with diameter $\mathrm{diam}(G)$. By the definition of diameter, there exists a path graph $P_n \subseteq G$ with $n = \mathrm{diam}(G)$. Therefore, a lower bound for the approximate graph distance problem on P_n implies a lower bound for the same problem on G.

The vertices of P_n can be listed in the order $v_1, v_2, ..., v_n$ such that the edges are (v_i, v_{i+1}), where $i = 1, 2, ..., n-1$. A given vertex v_i can then be labeled by a binary string $x_i \in \{0,1\}^m$, with $m = \Theta(\log n)$, and hence, given $v_i, v_j \in G$, $d_G(v_i, v_j) = |x_i - x_j|$. Therefore, a communication protocol which outputs $\widetilde{d}$ such that $(1-\epsilon)d_G(v_i,v_j) \leq \widetilde{d} \leq (1+\epsilon)d_G(v_i,v_j)$ is equivalent to a communication protocol which solves MOD_ϵ on inputs x_i, x_j. So computing an approximate modulus reduces to computing an approximate graph distance.

With this in mind, we can state our lower bound.

Theorem 2.3. *Given a graph G with diameter $\mathrm{diam}(G)$, the quantum communication complexity for the problem $\mathrm{DIS}_\epsilon[G]$ in the one-way communication model with $\epsilon < 1/4$ and failure probability 1/3 is $Q^1(\mathrm{DIS}_\epsilon[G]) = \Omega\big(\log \mathrm{diam}(G)\big)$.*

Proof. As mentioned before, the approximate graph distance problem on a path graph $P_n \subseteq G$ with $n = \mathrm{diam}(G)$ should be at least as hard as the same problem on G, i.e., $Q^1(\mathrm{DIS}_\epsilon[G]) \geq Q^1(\mathrm{DIS}_\epsilon[P_n])$. Moreover, $\mathrm{DIS}_\epsilon[P_n]$ is equivalent to MOD_ϵ on inputs of size $m = \Theta\big(\log \mathrm{diam}(G)\big)$, hence $Q^1(\mathrm{DIS}_\epsilon[G]) \geq Q^1(\mathrm{MOD}_\epsilon)$. According to Lemma 2.15, $Q^1(\mathrm{MOD}_\epsilon) = \Omega(Q^1(\mathrm{GT}))$, but $Q^1(\mathrm{GT}) = \Theta(m)$ [211, Appendix B], therefore $Q^1(\mathrm{DIS}_\epsilon[G]) = \Omega\big(\log \mathrm{diam}(G)\big)$. ■

The above result for the one-way communication model also holds for the SMP model. It then states that our communication protocol is optimal in terms of its dependence on $\mathrm{diam}(G)$.

2.4 Measuring ℓ_1-Distances

As seen in the previous sections, our communication protocol for approximating the Hamming distance can be adapted to ℓ_1-graphs. A graph G is said to be an ℓ_1-graph if there exist vectors $u_1, ..., u_n \in \mathbb{R}^m$ for some m, and with $n = |V(G)|$, such that $d_G(v_i, v_j) = \|u_i - u_j\|_1$ for all $v_i, v_j \in V(G)$. This connection between graphs and ℓ_1-norm suggests an application of our approximate Hamming distance protocol to ℓ_1-distances. More specifically, consider the following problem: Alice and Bob are each given a vector x, y (respectively) from $[-1, 1]^d$. Each entry of each vector is specified by k bits, for some k (1 bit to specify the sign, and $k-1$ bits $z_1, \ldots, z_{k-1}$ to specify a binary fraction $z_1 2^{-1} + z_2 2^{-2} + \cdots + z_{k-1} 2^{1-k}$). Alice and Bob's task is to approximate the ℓ_1-distance between x and y up to relative error ϵ in the SMP model.

A natural special case of this problem is where Alice and Bob are each given a probability distribution $x, y \in \mathbb{R}^d$, respectively, and are asked to approximately compute the ℓ_1-distance between them (equivalently, the total variation distance, which is defined as the ℓ_1-distance). This corresponds to the special case where $x_i, y_i \geq 0$ for all i, and $\sum_i x_i = \sum_i y_i = 1$.

Alice and Bob can use our approximate Hamming distance protocol to approximately compute $\|x - y\|_1$: the idea is to map these vectors into a Hamming metric via a unary representation [134]. Each entry $z \in [-1, 1]$ of each vector is mapped to a 2^k-bit string $s(z)$ such that the first $2^{k-1}(z+1)$ bits of $s(z)$ are set to 1, and the remaining bits are set to 0. Then, for any z, w, $|z - w| = d(s(z), s(w))/2^{k-1}$. Letting $s(x)$ denote the result of applying this map to each entry of x and concatenating the results, we have $\|x - y\|_1 = d(s(x), s(y))/2^{k-1}$ for bit strings $s(x)$, $s(y)$ of length $m = 2^k d$. So we can use our usual communication protocol (Protocol 2.7) to deliver an estimate of $\|x - y\|_1$ up to relative error ϵ using $O((\log^2 m)(\log\log m)/\epsilon^3)$ qubits of communication, which is $O((\log^2 d)(\log\log d)/\epsilon^3)$ when $k \leq \log d$.

The use of a unary representation may seem wasteful, but a straightforward binary representation would not preserve distances correctly for all inputs. There is also a lower bound that the communication complexity of this problem must have at least a linear dependence on k: by the lower bound on the complexity of the MOD_ϵ problem that follows from Lemma 2.15, $\Omega(k)$ bits of communication are required to approximately compute $\|x - y\|_1$ even for $d = 1$. Finally,

the protocol can easily be extended to the setting where $x, y \in [-M, M]^d$, for some $M \geq 1$, by rescaling the vectors appropriately.

2.A ℓ_1-Graphs Characterization

In this appendix we prove Theorem 2.11. Recall that $G(a|b) := \{x \in V(G) |\ d_G(x, a) < d_G(x, b)\}$.

Theorem 2.11. *A graph G is an ℓ_1-graph iff $G(a|b)$ is convex for each edge (a, b) of G.*

This theorem is a generalisation of *Djoković's Characterization* [62, 60] for partial cubes by introducing the concept of k-scale embedding, which is linked to the concept of ℓ_1-graphs. A partial cube is then just a special case of ℓ_1-graphs.

While the idea of k-scale embedding and some of its properties related to partial cubes were already studied, we could not find a clear and direct characterization for ℓ_1-graphs as it is stated in Theorem 2.11, similar to Djoković's. For example, in [181] it is proved that a graph is embeddable into a hypercube with an odd scale if and only if it is 1-embeddable into a hypercube, meaning that odd scale embeddings do not add anything new. This makes sense since an odd scale embedding cannot alter the bipartiteness requirement.

The proof of the theorem is sketched as follows. The direction $(i) \implies (ii)$ is a direct generalisation of Djoković's proof (see [60, Theorem 19.1.1]). On the other hand, the direction $(ii) \implies (i)$ does not follow Djoković's proof, but instead introduces the idea of a k-rescaling map which transforms a given connected and unweighted graph into a new graph by adding $k - 1$ new vertices on each original edge. In this way, the original distances are rescaled by a factor of k. We show in Lemma 2.22 that if k is even, then this new graph is bipartite. Also, we show in Lemma 2.24 that this map preserves the convexity of subgraphs. This means that, if the sets $G(a|b)$ are convex for each edge (a, b), then the new rescaled graph will fulfill the requirements from Djoković's Characterization for k even and is, therefore, a partial cube. Since the original vertices are a subset of the new ones, the original graph is an ℓ_1-graph.

In all the following, let $G = (V, E)$ be a connected and unweighted graph. We start by proving $(i) \implies (ii)$.

Lemma 2.18. *If G is an ℓ_1-graph, then $G(a|b)$ is convex for each edge (a, b) of G.*

Proof. Let (a, b) be an edge of G, let $x, y \in G(a|b)$ and $z \in V(G)$ lying on a shortest path from x to y. Consider a hypercube k-scale embedding $\sigma_k : V \to Q_n$ in which node a is labeled by $\sigma_k(a) = 0^n$ (where $c^j = ccc \cdots c$ means c repeated j times), node b is labeled by $\sigma_k(b) = 1^k 0^{n-k}$ and nodes x, y, z are labeled by the strings X, Y, Z. Given an n-bit string A, we define its i-th bit as A_i.

We first prove that $v \in G(a|b)$ if and only if $[\sigma_k(v)]_i \neq 1$ for $i \in [1, k]$. Consider that $[\sigma_k(v)]_i \neq 1$ for $i \in [1, k]$. Therefore $d_{Q_n}(\sigma_k(v), \sigma_k(b)) = k + d_{Q_n}(\sigma_k(v), \sigma_k(a))$ and hence v is closer to a

than b, i.e., $v \in G(a|b)$. Now consider that $v \in G(a|b)$. This means $d_{Q_n}(\sigma_k(v), \sigma_k(a)) = lk$ and $d_{Q_n}(\sigma_k(v), \sigma_k(b)) = (l+1)k$ for some $l \in \mathbb{N}$. Suppose that $[\sigma_k(v)]_i = 1$ for m indices i in $[1, k]$. Therefore $d_{Q_n}(\sigma_k(v), \sigma_k(b)) - k + m = d_{Q_n}(\sigma_k(v), \sigma_k(a)) - m$, which gives $(l+1)k - k + m = lk - m \implies m = 0$, i.e., $[\sigma_k(v)]_i \neq 1$ for $i \in [1, k]$.

Given this, then $X_i, Y_i \neq 1$ for $i \in [1, k]$, and $d_{Q_n}(X, Y) = d_{Q_n}(X, Z) + d_{Q_n}(Z, Y)$ since $d_G(x, y) = d_G(x, z) + d_G(z, y)$. This implies that $Z_i \neq 1$ for $i \in [1, k]$, i.e., $z \in G(a|b)$. This shows that the set $G(a|b)$ is convex. ■

To prove $(ii) \implies (i)$, we first make the following definitions.

Definition 2.19. Given $G = (V, E)$, let $\mathbb{G}_k : G \to G^{(k)}$ be the k-rescaling map which adds $k-1$ new nodes on every edge $e \in E$. The resulting graph $G^{(k)} = (V^{(k)}, E^{(k)})$ is called the k-rescaled image of G. Also, $G^{(1)} = G$. It is straightforward that $|E^{(k)}| = k|E|$ and $|V^{(k)}| = |V| + (k-1)|E|$. Moreover, given $S \subseteq V$, we shall write $\mathbb{G}_k(S)$ for the vertex set of $\mathbb{G}_k(G[S])$, where $G[S]$ is the subgraph induced by S.

Definition 2.20. Let $v \in V$. We define $G \oplus (v, v')$ as the graph $G' = (V', E')$ obtained by connecting an extra node v' to the node v, so that $V' = V \cup \{v'\}$ and $E' = E \cup (v, v')$. If $v' = v$, we define $G \oplus (v, v) = G$.

Definition 2.21. Let $\mathbb{G}_k : G \to G^{(k)}$. Given $(u, v) \in E$, we define the set $V^{(k)}(u, v) := \{w \in V^{(k)} |\ d_{G^{(k)}}(u, w) < k \text{ and } d_{G^{(k)}}(v, w) < k\}$.

The set $V^{(k)}(u, v)$ is just the nodes added between the nodes $u, v \in V$. With the above definition, $V^{(k)} = V \cup \left(\bigcup_{e \in E} V^{(k)}(e)\right)$.

We now state the following auxiliary lemmas.

Lemma 2.22. *The k-rescaled image $G^{(k)}$ of G is bipartite if k is even.*

Proof. A graph G is bipartite if and only if it does not contain an odd cycle. If G does not have cycles, then neither does $G^{(k)}$, since the k-rescaling map $\mathbb{G}_k$ cannot create cycles. Therefore $G^{(k)}$ is bipartite. Now suppose G has cycles. Given a cycle $S \subseteq V$, its k-rescaled image $S^{(k)} = \mathbb{G}_k(S)$ is such that $|S^{(k)}| = k|S|$. If S is an even cycle, then so is $S^{(k)}$. If S is an odd cycle, then $S^{(k)}$ is an even cycle if k is even. Therefore $S^{(k)}$ cannot have odd cycles for k even and hence is bipartite. ■

The next lemma says that augmenting a graph with a new vertex connected via a new edge preserves convexity.

Lemma 2.23. *Let $S \subseteq V$ and $v \in S$. Construct the new augmented graph $G' = G \oplus (v, v')$ for some $v' \notin V$ and consider the new subset $S' = S \cup \{v'\}$. If S is convex, then so is S'.*

Proof. Let $x, y \in S'$ and $z \in V \cup \{v'\}$ be such that $d_{G'}(x,y) = d_{G'}(x,z) + d_{G'}(y,z)$. We have two cases: Either $x, y \in S$ or, without loss of generality, $x = v'$ and $y \in S$. If $x, y \in S$, then it is straightforward that $z \neq v'$, otherwise the node $v \in S$ would be traversed twice. Therefore $z \in V$ and since S is convex, $z \in S \subset S'$ and S' is convex. On the other hand, if $x = v'$ and $y \in S$, the fact that v' is only connected to v implies that $d_{G'}(v', y) = d_{G'}(v', z) + d_{G'}(y, z) \iff d_G(v, y) = d_G(v, z) + d_G(y, z)$, which, together with S being convex, means that $z \in S \subset S'$. ■

Lemma 2.24. *Let $G^{(k)} = (V^{(k)}, E^{(k)})$ be the k-rescaled image of G. Then $S \subseteq V$ is convex if and only if $S^{(k)} := \mathbb{G}_k(S) \subseteq V^{(k)}$ is convex.*

Proof. We start by proving S convex $\implies$ $S^{(k)}$ convex. Let $x, y \in S^{(k)}$ and $z \in V^{(k)}$ be such that $d_{G^{(k)}}(x,y) = d_{G^{(k)}}(x,z) + d_{G^{(k)}}(z,y)$. We will show that $z \in S^{(k)}$. Let us define the sets $A = \{a \in V |\ d_{G^{(k)}}(x,y) = d_{G^{(k)}}(x,a) + d_{G^{(k)}}(a,y)\}$ and $A' = \{a \in V^{(k)} \setminus V |\ d_{G^{(k)}}(x,y) = d_{G^{(k)}}(x,a) + d_{G^{(k)}}(a,y)\}$, i.e., A is the set of original nodes that lie in the shortest path between x and y, while A' is the set of added nodes that lie in the shortest path between x and y. Note that $z \in A \cup A'$. Suppose $A = \emptyset$. This means that $x, y \in S^{(k)}(e)$ for some edge $e \in E$. Therefore we must have $z \in S^{(k)}(e) \subseteq S^{(k)}$.

Now suppose $A \neq \emptyset$. Let $a(x), a(y) \in A$ be the closest nodes to x and y, respectively. We must have $a(x) \in S$ (and $a(y) \in S$) since either $x \in S$ and then $a(x) = x$, or $x \in S^{(k)}(e)$ for some edge e, and then $a(x)$ is an endnode of e. We can have two situations: either $a(x) = a(y)$ or $a(x) \neq a(y)$.

Suppose $a(x) = a(y)$. Since $x \neq y$, this means that $x \in S^{(k)}(a(x), v_1)$ and $y \in S^{(k)}(a(x), v_2)$, for some $v_1, v_2 \in V$ such that $v_1 \neq v_2$, i.e., they are added nodes to two different edges with the common node $a(x)$. Therefore either $z \in S^{(k)}(a(x), v_1)$ or $z \in S^{(k)}(a(x), v_2)$ or $z = a(x)$, which lead to $z \in S^{(k)}$.

Suppose then that $a(x) \neq a(y)$. Consider for now that $z \in A$. Since S is convex and $d_{G^{(k)}}(a(x), a(y)) = d_{G^{(k)}}(a(x), z) + d_{G^{(k)}}(a(y), z) \iff d_G(a(x), a(y)) = d_G(a(x), z) + d_G(a(y), z)$, we conclude that $z \in S$, i.e., $A \subseteq S$. Now consider that $z \in A'$, so $z \in V^{(k)}(v_1, v_2)$ for some nodes $v_1, v_2 \in V$. We must have $v_1, v_2 \in S$. Indeed, if $x \in V^{(k)}(v_1, v_2)$ (or y), by construction $x \in S^{(k)}$ and $x \in V^{(k)}(v_1, v_2) \implies v_1, v_2 \in S$. And if $x, y \notin V^{(k)}(v_1, v_2)$, it implies that $v_1, v_2 \in A \subseteq S$. Hence $z \in S^{(k)}(v_1, v_2) \subseteq S^{(k)}$. Thus $z \in S^{(k)}$ in all cases, so $S^{(k)}$ is convex.

We now prove the other direction, $S^{(k)}$ convex $\implies$ S convex. Let $x, y \in S$ and $z \in V$ be such that $d_G(x,y) = d_G(x,z) + d_G(z,y)$. Suppose $z \notin S$. Remembering the equivalence between $d_{G^{(k)}}$ and d_G, this implies that $\exists z \in V \subset V^{(k)}$ but $z \notin S^{(k)}$ such that $d_{G^{(k)}}(x,y) = d_{G^{(k)}}(x,z) + d_{G^{(k)}}(z,y)$ for $x, y \in S \subset S^{(k)}$, which is a contradiction since $S^{(k)}$ is convex. We conclude that $z \in S$ and S is convex. ■

The above lemmas lead to the following one.

Lemma 2.25. *Let $G^{(k)} = (V^{(k)}, E^{(k)})$ be the k-rescaled image of $G = (V, E)$. If $G(a|b)$ is convex for each $(a, b) \in E$, then $G^{(k)}(a'|b')$ is also convex for each $(a', b') \in E^{(k)}$.*

Proof. Consider the edge $(a', b') \in E^{(k)}$ such that $a', b' \in \{a, b\} \cup V^{(k)}(a, b)$ for $a, b \in V$, i.e., $(a, b) \in E$ is the original edge. We note that the subgraph induced by $G^{(k)}(a'|b')$ is just $\mathbb{G}_k\big(G[G(a|b)]\big) \oplus (a, w_1) \oplus (w_1, w_2) \oplus \cdots \oplus (w_n, a')$ for $n := d_{G^{(k)}}(a, a') - 1$ (if $n = 0$, then $\mathbb{G}_k\big(G[G(a|b)]\big) \oplus (a, a')$, and if $a = a'$, then just $\mathbb{G}_k\big(G[G(a|b)]\big)$). Since $G(a|b)$ is convex, by Lemma 2.24 $\mathbb{G}_k\big(G(a|b)\big)$ is also convex, and finally by Lemma 2.23 $G^{(k)}(a'|b')$ is convex. ■

Finally, with the above lemmas, we can prove $(ii) \implies (i)$ in Theorem 2.11.

Lemma 2.26. *If $G(a|b)$ is convex for each edge (a, b) of $G = (V, E)$, then G is an ℓ_1-graph.*

Proof. Consider the k-rescaled graph $G^{(k)} = (V^{(k)}, E^{(k)})$ corresponding to G for k even. By Lemma 2.22, $G^{(k)}$ is bipartite. By Lemma 2.25, $G^{(k)}(a'|b')$ is convex for each $(a', b') \in E^{(k)}$. Therefore, by Djoković's characterization $G^{(k)}$ can be isometrically embedded into a hypercube [62]. Since $V \subset V^{(k)}$, we conclude that G can be k-embedded into the same hypercube, i.e., it is an ℓ_1-graph. ■

Chapter 3

Generalisations of the Boolean Hidden Matching Problem

The Hidden Matching problem [20] was the first problem to exhibit an exponential separation between the bounded-error classical communication complexity and the bounded-error quantum communication complexity in the one-way model. The problem can be efficiently solved by one quantum message of $\log n$ qubits, while any classical one-way protocol needs to send $\Theta(\sqrt{n})$ bits to solve it. The hardness of the problem is essentially one-way: it could be efficiently solved by having Bob sent a classical message of $\log n$ bits to Alice after Alice's message. The Hidden Matching problem is a relational problem. In the same paper [20], Bar-Yossef, Jayram and Kerenidis proposed a Boolean version of the problem, the Boolean Hidden Matching problem (which is a partial Boolean function), and conjectured that the same quantum-classical gap holds for it as well, which was later proven to be true by Gavinsky, Kempe, Kerenidis, Raz and de Wolf [81]. In this chapter we generalise this separation.

3.1 Previous Work on Hidden Matching problems

In what follows, $[n] := \{1, 2, \ldots, n\}$, $\mathbb{S}_n$ is the set of permutations of $[n]$ and given $x, y \in \{-1, 1\}^n$, we denote by $x \circ y$ the Hadamard (elementwise) product of x and y, and by $\overline{x}$ the complement of x, such that $x \circ \overline{x} = -1^n$. Moreover, given an expression $\mathcal{E}$, we denote by $\mathbf{1}[\mathcal{E}]$ the Iverson bracket, i.e., indicator function $\mathbf{1}[\mathcal{E}] = 1$ if $\mathcal{E}$ is true and 0 if not.

The Hidden Matching (HM_n^α) and Boolean Hidden Matching (BHM_n^α) problems are defined with respect to some $\alpha \in (0, 1]$. Alice is given a string $x \in \{-1, 1\}^n$ and Bob is given a sequence $M \in \mathcal{M}_{\alpha n/2}$ of $\alpha n/2$ disjoint pairs $(i_1, j_1), (i_2, j_2), \ldots, (i_{\alpha n/2}, j_{\alpha n/2}) \in [n]^2$. Such a sequence is called an α-matching, and $\mathcal{M}_{\alpha n/2}$ denotes the family of all α-matchings – partial matchings of a

fixed size in the complete graph on n vertices. Together x and M induce a string $z \in \{-1,1\}^{\alpha n/2}$ defined by the parities of the $\alpha n/2$ edges, i.e., $z_\ell = x_{i_\ell} x_{j_\ell}$ for $\ell = 1, \ldots, \alpha n/2$. Then the HM_n^α and BHM_n^α problems are defined as follows.

The Hidden Matching problem (HM_n^α)**.** Let $n \in \mathbb{N}$ be even and $\alpha \in (0,1]$. Alice receives $x \in \{-1,1\}^n$ and Bob receives $M \in \mathcal{M}_{\alpha n/2}$. Their goal is to output a tuple $\langle i,j,b \rangle$ such that $(i,j) \in M$ and $b = x_i x_j$.

The Boolean Hidden Matching problem (BHM_n^α)**.** Let $n \in \mathbb{N}$ be even and $\alpha \in (0,1]$. Alice receives $x \in \{-1,1\}^n$ and Bob receives $M \in \mathcal{M}_{\alpha n/2}$ and $w \in \{-1,1\}^{\alpha n/2}$. It is promised that $z \circ w = b^{\alpha n/2}$ for some $b \in \{-1,1\}$. Their goal is to output b.

Given inputs x and M, it is clear that there are many possible correct outputs for the HM_n^α problem ($\alpha n/2$ correct outputs, actually), making it a relational problem. On the other hand, the BHM_n^α is a partial Boolean function due to the promise statement.

Bar-Yossef *et al.* [20] gave a simple quantum protocol to solve the HM_n^1 problem with just $O(\log n)$ qubits of communication[1], while proving that any classical protocol needs to communicate $\Theta(\sqrt{n})$ bits. While the lower bound is technical and relies on information theory, the upper bound comes from a simple protocol: Alice sends c of her bits (and their position) uniformly at random to Bob. If $c = O(\sqrt{n})$, then, by a Birthday Paradox argument, at least two of her bits will fall into the same edge of Bob's matching with high probability. Indeed, the probability that at least one of the $n/2$ edges is filled by two bits is $1 - 2^c \binom{n/2}{c} / \binom{n}{c} = 1 - \Theta\Big(\frac{(1-c/n)^{n-c+1/2}}{(1-2c/n)^{n/2-c+1/2}}\Big) = 1 - \Theta(e^{-c^2/n + c/2n})$ by using Stirling's approximation $n! = \Theta(n^{n+1/2} e^{-n})$. By taking $c = \Theta(\sqrt{n})$, this probability is $1 - \delta$ for small δ, meaning that HM_n^1 can be solved with high probability with $O(\sqrt{n} \log n)$ bits, which can be further reduced to $O(\sqrt{n} + \log n)$ via Newman's theorem [152].

Similarly with the BHM_n^α problem, Gavinsky *et al.* [81] demonstrated the same exponential classical-quantum communication gap for any $\alpha \leq 1/2$ (note that their definition of α differs from ours by a factor of 2). As HM_n^α is at least as difficult as BHM_n^α, their result implies the same lower bound for HM_n^α. The approach taken by Gavinsky *et al.* in proving the classical lower bound is particularly interesting in that it uses the KKL inequality [112].

A slightly weaker separation ($O(\log n)$ vs. $\Omega(n^{7/16})$) for a closely related problem was shown in [145] using similar techniques. The BHM_n^α problem was generalised by Verbin and Yu [192] to a problem that they named Boolean Hidden Hypermatching (BHH_n^t). In this problem, instead of having the bits from Alice matched in pairs, they are now matched in tuples of t elements. In other words, a bit from the final string z is obtained by XORing t bits from Alice's string. More precisely, Alice is given a string $x \in \{-1,1\}^n$ and Bob is given a sequence $M \in \mathcal{M}_{n/t}$ of n/t disjoint tuples $(M_{1,1}, \ldots, M_{1,t}), \ldots, (M_{n/t,1}, \ldots, M_{n/t,t}) \in [n]^t$ called a hypermatching, where $\mathcal{M}_{n/t}$ denotes the family of all hypermatchings. Both x and M induce a string $z \in \{-1,1\}^{n/t}$ defined by the parities of the n/t edges, i.e., $z_\ell = \prod_{j=1}^t x_{M_{\ell,j}}$ for $\ell = 1, \ldots, n/t$. The BHH_n^t problem is defined as follows.

[1] Their protocol extends easily to the more general HM_n^α problem.

The Boolean Hidden Hypermatching problem **(BHH$_n^t$).** Let $n, t \in \mathbb{N}$ be such that $2t|n$. Alice receives $x \in \{-1,1\}^n$ and Bob receives $M \in \mathcal{M}_{n/t}$ and $w \in \{-1,1\}^{n/t}$. It is promised that $z \circ w = b^{n/t}$ for some $b \in \{-1,1\}$. Their goal is to output b.

Verbin and Yu proved a classical lower bound of $\Omega(n^{1-1/t})$ communication for every bounded-error one-way protocol, showing the increasing hardness of the problem with t, as one should expect since the BHH$_n^t$ problem can be reduced from the BHM$_n$ problem (we will show how this is done in detail later). The authors subsequently used this problem to prove various streaming lower bounds, i.e., lower bounds on the space required of streaming algorithms (algorithms that read the input from left to right, use a small amount of space, and approximate some function of the input). However, no efficient quantum protocol was proposed for solving the BHH$_n^t$ problem for $t > 2$. It was only later that Shi, Wu and Yu [177] showed that such efficient quantum protocols do not exist. More specifically, they proved a quantum lower bound of $\Omega(n^{1-2/t})$ communication for every bounded-error one-way protocol for the BHH$_n^t$ problem. Their proof is similar to the ones used in the classical lower bound, the difference lying in the use of Fourier analysis of *matrix-valued* functions and the matrix-valued Hypercontractive Inequality of Ben-Aroya, Regev and de Wolf [24].

Note that the lower bound of Verbin and Yu does not use an α parameter, unlike the lower bound of [81]. However, their lower bound requires n/t to be even, otherwise Alice can just send the parity of her bit-string. (The result of [81] can be extended to hold for any $\alpha < 1$ fairly straightforwardly, but achieving a strong lower bound for $\alpha = 1$ requires some more work.)

3.1.1 Our Results

This chapter focuses on the study of a broad generalisation of the BHH$_n^t$ problem. In the (Boolean) Hidden Matching and Boolean Hidden Hypermatching problems, the task Alice and Bob want to solve can be viewed as rearranging Alice's data according to some permutation that Bob holds, and 'compressing' the data to a final bit-string by applying some Boolean function to the bits. Then Alice and Bob's goal is to determine some information about this final bit-string. The way this compression was originally done was via the Parity function, but, apart from the obvious reason that Parity gives the desired classical-quantum communication gap and, less obviously, leads to a clear proof, there is no particular need to restrict to this function in order to arrive at the final bit-string. This observation leads to a generalisation of the Boolean Hidden Hypermatching problem, which we named the f-Boolean Hidden Partition (f-BHP$_n^{\alpha,t}$) problem, where $f : \{-1,1\}^t \to \{-1,1\}$ is the Boolean function used to compress Alice's bits.

Given $y \in \{-1,1\}^n$, we define by $y^{(j)} = (y_{(j-1)t+1}, y_{(j-1)t+2}, \ldots, y_{jt}) \in \{-1,1\}^t$ the j-th block of size t from y, with $t|n$ and $j = 1, \ldots, n/t$. The f-Boolean Hidden Partition problem is defined as follows. Alice is given a bit-string $x \in \{-1,1\}^n$, and Bob is given a permutation $\sigma \in \mathbb{S}_n$ and a bit-string $w \in \{-1,1\}^{\alpha n/t}$, where $\alpha \in (0,1]$ is fixed and $t|n$. Given a Boolean function $f : \{-1,1\}^t \to \{-1,1\}$, we can define the map $B_f : \{-1,1\}^n \times \mathbb{S}_n \to \{-1,1\}^{\alpha n/t}$

by $B_f(x,\sigma) = \left(f(\sigma(x)^{(1)}),\ldots,f(\sigma(x)^{(\alpha n/t)})\right)$, where $\sigma(x)_i = x_{\sigma^{-1}(i)}$. Hence x and σ induce a bit-string given by $B_f(x,\sigma)$, each of whose bits is obtained by applying f to a block of the permuted bit-string $\sigma(x)$. The f-BHP$_n^{\alpha,t}$ problem can be defined as follows.

The f-Boolean Hidden Partition problem **(f-BHP$_n^{\alpha,t}$).** Let $n,t \in \mathbb{N}$ be such that $t|n$ and $\alpha \in (0,1]$. Alice receives $x \in \{-1,1\}^n$ and Bob receives $\sigma \in \mathbb{S}_n$ and $w \in \{-1,1\}^{\alpha n/t}$. It is promised that there is $b \in \{-1,1\}$ such that $B_f(x,\sigma) \circ w = b^{\alpha n/t}$. The problem is to output b.

The adoption of the word 'Partition' instead of '(Hyper)Matching' from previous works comes from our decision to view the problem in terms of a hidden partition that Bob holds, instead of an α-(Hyper)Matching. Bob shuffles Alice's data according to some permutation, and then just partitions the resulting data in adjacent blocks of size t and uses f to get the final bit-string. Obviously both views are equivalent, but we think that the permutation approach eases the analysis of the problem.

Our aim is to study the f-Boolean Hidden Partition problem in terms of the function f. It should be clear that for some functions the problem is hard to solve classically, e.g. when f is the Parity function and we recover the usual Boolean Hidden Hypermatching problem. On the other hand, for some functions it becomes easily solvable, e.g. when f is the AND function (Alice needs only to send the position of any 0 in her string). We would like to characterize for which functions the problem can be efficiently solved classically, i.e., with $O(\log n)$ bits of communication, and for which functions it is hard to solve classically, i.e., requires $\Omega(n^a)$ bits of communication for some $a \in (0,1]$. And the same question applies to quantum communication complexity: we would like to determine for which functions the problem admits or not an efficient quantum communication protocol. Given this characterization, we can check for which functions there is an exponential classical-quantum communication gap.

We conjecture that the whole f-BHP$_n^{\alpha,t}$ problem can be characterized mainly by the *sign-degree* of the function f, and we give substantial evidence for such conjecture. A polynomial $p : \{-1,1\}^t \to \mathbb{R}$ is said to *sign-represent* f if $f(x) = \mathrm{sgn}(p(x))$. If $|p(x)| \leq 1$ for all x, we say that p is *normalized*. The *bias* of a normalized polynomial p is defined as $\beta = \min_x |p(x)|$. The sign-degree ($\mathrm{sdeg}(f)$) of f is the minimum degree of polynomials that sign-represent it. In Section 3.2 we give upper bounds on the classical and quantum communication complexity of the f-Boolean Hidden Partition problem based on the sign-degree.

Theorem 3.1+3.4. *Let $f : \{-1,1\}^t \to \{-1,1\}$ be a Boolean function. If $\mathrm{sdeg}(f) \leq 1$, then there exists a bounded-error classical protocol that solves the f-BHP$_n^{\alpha,t}$ problem with error probability ϵ and $O\left((\frac{t}{\alpha\beta})^2 \log\frac{1}{\epsilon}\log n\right)$ bits of communication. If $\mathrm{sdeg}(f) \leq 2$, then there exists a bounded-error quantum protocol that solves the f-BHP$_n^{\alpha,t}$ problem with error probability ϵ and $O\left((\frac{t}{\alpha\beta})^2 \log\frac{1}{\epsilon}\log n\right)$ qubits of communication. In both these results, β is the bias of any normalized polynomial of degree $\mathrm{sdeg}(f)$ that sign-represents f.*

Note that the bias β can be very small, but can also be lower-bounded in terms of t and $\mathrm{sdeg}(f)$: indeed, it is shown in [41] that β is lower-bounded by $t^{-O(t^{\mathrm{sdeg}(f)})}$. In this work we will usually assume that $t = O(1)$, so $\beta = \Omega(1)$. We assume throughout that Alice and Bob do not have access to shared randomness or entanglement. The classical complexity in the above theorem can actually be improved to an additive dependence on $\log n$ via applying Newman's Theorem [152] to a protocol with shared randomness, but at the expense of making the protocol less intuitive.

The classical upper bound stated above comes from the observation that, if f has a sign-representing polynomial p of degree 1, it is possible to determine whether $f(z) = 1$ with probability $> 1/2$ by only evaluating f on one uniformly random bit of z, by writing down a probabilistic procedure whose expectation on z mimics $p(z)$. So Alice sends a few uniformly random bits to Bob, who matches them to blocks in his partition, and evaluates f on the corresponding blocks with success probability $> 1/2$ for each block. Only a few repetitions are required to determine whether $f(x) = w$ or $f(x) = \overline{w}$ with high probability.

On the other hand, to obtain the quantum upper bound we use the idea of *block-multilinear* polynomials from [2, 3], and some auxiliary results also from [3]. The idea is that Alice sends a superposition of her bits, and Bob, after collapsing the state onto one of the blocks from his partition (say block j), applies a controlled unitary operator that describes a block-multilinear polynomial $\widetilde{p}$ of degree 2, which is produced from a sign-representing polynomial p for f of degree 2. A Hadamard test is used to return an output with probability depending (roughly speaking) on $\widetilde{p}(\sigma(x)^{(j)}, \sigma(x)^{(j)})$, which in turn is equal to $p(\sigma(x)^{(j)})$ according to a theorem from [3]. The Hadamard test then outputs 1 with probability greater than $1/2$ if $f(x^{(j)}) = 1$ and 0 with probability greater than $1/2$ if $f(x^{(j)}) = -1$.

We remark that both of these protocols actually solve a natural generalisation of the Hidden Matching problem [20] (i.e., they output the result of evaluating $f(x^{(j)})$ for Bob's block j, where j is arbitrary), which is at least as hard as the f-Boolean Hidden Partition problem. However, unlike the Hidden Matching problem, the output is not correct with certainty, but only with probability strictly greater than $1/2$.

In Sections 3.3, 3.4 and 3.5 we prove classical and quantum lower bounds. In Section 3.3 we reduce the Boolean Hidden Matching problem to the f-Boolean Hidden Partition problem and prove that for almost all symmetric Boolean function f with $\mathrm{sdeg}(f) \geq 2$ the f-$\mathrm{BHP}_n^{\alpha,t}$ problem requires at least $\Omega(\sqrt{n})$ bits of communication. The only functions for which the reduction does not work are the Not All Equal functions on an odd number of bits, i.e., $\mathrm{NAE} : \{-1,1\}^t \to \{-1,1\}$, defined by $\mathrm{NAE}(x) = -1$ if $|x| \in \{0, t\}$ and $\mathrm{NAE}(x) = 1$ otherwise, with t odd.

Theorem 3.7. *Let $f : \{-1,1\}^t \to \{-1,1\}$ be a symmetric Boolean function with $\mathrm{sdeg}(f) \geq 2$. If f is not the NAE function on an odd number of bits, then any bounded-error classical communication protocol for the f-$\mathrm{BHP}_n^{\alpha,t}$ problem needs to communicate $\Omega(\sqrt{n/(\alpha t)})$ bits.*

Finally, in Sections 3.4 and 3.5 we generalise the Fourier analysis methods from [81, 192, 177] to prove a partial result on the hardness of the f-BHP$_n^{\alpha,t}$ problem, both classically and quantumly. Ideally we would like to prove that any bounded-error classical and quantum protocols would need to communicate $\Omega(n^{1-1/d})$ bits and $\Omega(n^{1-2/d})$ qubits, respectively, where $\mathrm{sdeg}(f) = d$. What we obtained is this result but with d being the *pure high degree* of f. A Boolean function f is said to have pure high degree ($\mathrm{phdeg}(f)$) d if $\widehat{f}(S) = 0$ for all $|S| = 0, 1, \ldots, d-1$. It is possible to prove that $\mathrm{phdeg}(f) \leq \mathrm{sdeg}(f)$ [176] (see also [43, Theorem 1]), so our result is a step towards proving a lower bound for all functions with sign degree ≥ 2.

Theorem 3.8 + 3.11. *Let $f : \{-1,1\}^t \to \{-1,1\}$ be a Boolean function. If $\mathrm{phdeg}(f) = d \geq 2$, then, for sufficiently small $\alpha > 0$ that does not depend on n, any bounded-error classical communication protocol for solving the f-BHP$_n^{\alpha,t}$ problem needs to communicate at least $\Omega(n^{1-1/d})$ bits. If $\mathrm{phdeg}(f) = d \geq 3$, then, for sufficiently small $\alpha > 0$ that does not depend on n, any bounded-error quantum communication protocol for solving the f-BHP$_n^{\alpha,t}$ problem needs to communicate at least $\Omega(n^{1-2/d})$ qubits.*

The classical proof in Section 3.4 follows the general idea from [81, 192], but the technical execution was substantially changed by borrowing ideas from [177]. First, we apply Yao's minimax principle (see Theorem 1.11). Alice sends a message to Bob. If the length of the message sent is c, then the inputs for which Alice could have sent that specific message define a set A of about 2^{n-c} x's. From Bob's perspective, he knows that the random variable X corresponding to Alice's bit-string is uniformly distributed in a set A and he knows his permutation σ, hence his knowledge of the random variable $B_f(X, \sigma)$ is described by the distributions

$$p_\sigma(z) = \frac{|\{x \in A | B_f(x,\sigma) = z\}|}{|A|} \text{ and } q_\sigma(z) = \frac{|\{x \in A | B_f(x,\sigma) = \overline{z}\}|}{|A|}. \tag{3.1}$$

It is well known that the best success probability for distinguishing two distributions q_1 and q_2 with one sample is $1/2 + \|q_1 - q_2\|_{\mathrm{tvd}}/4$, where $\|q_1 - q_2\|_{\mathrm{tvd}} := \sum_i |q_1(i) - q_2(i)|$. Therefore the bias of the protocol, i.e., the protocol's successful probability minus a half, is equal to $\|p_\sigma - q_\sigma\|_{\mathrm{tvd}}/4$. Differently from the approach of [81, 192], and following [177], we directly upper bound the expectation of the bias over Bob's permutation. By demanding a small distributional error, we arrive at the desired communication lower bound. Upper bounding the bias is done via Fourier analysis, using the KKL inequality.

The quantum proof in Section 3.5 follows the same idea from [177]. Yao's minimax principle is still applied, and the best strategy for Bob in determining b conditioned on his input (σ, w) is no more than the chance to distinguish between two subsets of Alice's messages, where a message corresponds to a quantum state ρ_x, selected according to b. It is known that any protocol that tries to distinguish two quantum states ρ_0 and ρ_1 appearing with probability p and $1-p$, respectively, by a POVM has bias at most $\|p\rho_0 - (1-p)\rho_1\|_{\mathrm{tr}}/2$ [98]. The bias is

then upper bounded by using Fourier analysis of matrix-valued functions, in particular by the matrix-valued hypercontractive inequality of Ben-Aroya, Regev and de Wolf [24].

The difference between the classical and quantum lower bound proofs was considerably reduced in our procedure, e.g. the need for Parseval's identity is substantially reduced in the classical proof. Still some differences persist. Apart from the obvious generalisation of Fourier analysis to matrix-valued functions, the Fourier analysis in the quantum lower bound proof is performed directly on the encoding messages and not on the pre-images of a fixed encoding message, since there is no clear quantum analogue of conditioning on a message. The main technical difficulty we faced compared to [81, 192] is that the Fourier coefficients of Bob's distributions $p_\sigma(z)$ and $q_\sigma(z)$ are not nicely related to just one Fourier coefficient of the characteristic function of A any more, but instead to a more complicated sum of many coefficients. This requires us to carefully bound various combinatorial terms occurring in the proof and to use our freedom to choose α fairly small.

In Section 3.6 we analyse the limitations of our techniques and show that under the uniform distribution, which was used as the 'hard' distribution during the proof of Theorem 3.8, we cannot obtain a lower bound depending on the sign degree instead of the pure high degree.

We finally remark that the one-way communication complexity separations we found can easily be used to obtain corresponding separations in the streaming model, similarly to [81, 192].

3.2 Classical and Quantum Upper Bounds

The sign-representing polynomial p allows us to build efficient classical and quantum communication protocols depending on $\mathrm{sdeg}(f)$. We shall show that there is an efficient $O(\log n)$-bit classical communication protocol for solving the f-$\mathrm{BHP}_n^{\alpha,t}$ problem if $\mathrm{sdeg}(f) \leq 1$. On the other hand, we shall show that there is an efficient $O(\log n)$ quantum communication protocol for solving the f-$\mathrm{BHP}_n^{\alpha,t}$ problem if $\mathrm{sdeg}(f) \leq 2$.

Intuitively, the contrast between $\mathrm{sdeg}(f) \leq 1$ for the classical protocols and $\mathrm{sdeg}(f) \leq 2$ for the quantum protocols comes from the nature of probability distributions in each case. One wants to relate the probability of outputting the right answer with the sign-representing polynomial p: if $p(x) > 0$, we would like to output 1 with high probability, and if $p(x) < 0$, we would like to output 1 with low probability. Classically, this probability distribution can only depend linearly on the bits of x, but quantumly, since this probability distribution arises from the square of a quantum amplitude, it can have a quadratic dependence on the bits of x.

3.2.1 Classical Upper Bound

Consider the f-BHP$_n^{\alpha,t}$ problem for $f : \{-1,1\}^t \to \{-1,1\}$ with $\mathrm{sdeg}(f) \leq 1$. Let $p : \{-1,1\}^t \to [-1,1]$ be a normalized sign-representing polynomial for f. Hence we can write

$$p(x) = \alpha_0 + \sum_{i=1}^{t} \alpha_i x_i \tag{3.2}$$

with $(\alpha_i)_{i=0}^t \in \mathbb{R}$. Let $\beta = \min_x |p(x)|$ be the bias of p.

Theorem 3.1. *$R_\epsilon^1(f\text{-BHP}_n^{\alpha,t}) = O\left((\frac{t}{\alpha\beta})^2 \log\frac{1}{\epsilon} \log n\right)$ if $\mathrm{sdeg}(f) \leq 1$, where β is the bias of any normalized sign-representing polynomial for f with degree ≤ 1.*

Proof. Consider the following protocol: Alice picks $m = O\big((\frac{t}{\alpha\beta})^2 \log\frac{1}{\epsilon}\big)$ bits from x uniformly at random (with replacement) and sends them to Bob, together with their indices. Let I and $\{x_i\}_{i\in I}$ be the indices and bitvalues sent, respectively. Let $j(i) = \lceil \sigma(i)/t \rceil$ and $k(i) \equiv \sigma(i) \mod t$ for all $i \in I$, where $\sigma \in \mathbb{S}_n$ is Bob's permutation. Define the random variable

$$X(i) := \begin{cases} (\alpha_{k(i)} x_i + \alpha_0/t) w_{j(i)} & \text{if } \sigma(i) \in [\alpha n/t], \\ 0 & \text{if } \sigma(i) \notin [\alpha n/t], \end{cases} \tag{3.3}$$

where α_0 and α_k are the zeroth order and x_k's coefficients, respectively, from the sign-representing polynomial p, and define $X := \sum_{i\in I} X(i)$. Bob then computes $\mathrm{sgn}(X)$. If the sign is 1, he outputs $B_f(x,\sigma) = w$, and if the sign is -1, he outputs $B_f(x,\sigma) = \overline{w}$.

To see why the protocol works, we calculate the expected value of the random variable X.

$$\begin{aligned}
\mathbb{E}[X] &= m \cdot \mathbb{E}_i[X(i)] && (3.4)\\
&= \alpha m \cdot \mathbb{E}_i[(\alpha_{k(i)} x_i + \alpha_0/t) w_{j(i)}] && (3.5)\\
&= \alpha m \cdot \mathbb{E}_j\left[\mathbb{E}_k[\alpha_k \sigma(x)_k^{(j)} + \alpha_0/t] w_j\right] && (3.6)\\
&= \alpha m \cdot \mathbb{E}_j\left[\frac{p(\sigma(x)^{(j)})}{t} w_j\right] && (3.7)\\
&= \alpha m \frac{t}{n} \sum_{j=1}^{n/t} \frac{p(\sigma(x)^{(j)})}{t} w_j && (3.8)\\
&= \frac{\alpha m}{n}\left[\sum_{j:w_j=1} p(\sigma(x)^{(j)}) - \sum_{j:w_j=-1} p(\sigma(x)^{(j)})\right]. && (3.9)
\end{aligned}$$

If $f(\sigma(x)^{(j)}) = w_j$, then $w_j = 1 \implies p(\sigma(x)^{(j)}) \geq \beta > 0$ and $w_j = -1 \implies p(\sigma(x)^{(j)}) \leq -\beta < 0$. Therefore

$$\mathbb{E}[X] \geq \frac{\alpha m}{n}\left[\sum_{j:w_j=1} \beta - \sum_{j:w_j=-1} -\beta\right] = \alpha m \frac{\beta}{t}. \tag{3.10}$$

If, on the other hand, $f(\sigma(x)^{(j)}) = -w_j$, then $w_j = 1 \implies p(\sigma(x)^{(j)}) \leq -\beta < 0$ and $w_j = -1 \implies p(\sigma(x)^{(j)}) \geq \beta > 0$. Therefore

$$\mathbb{E}[X] \leq \frac{\alpha m}{n}\left[\sum_{j:w_j=1} -\beta - \sum_{j:w_j=-1} \beta\right] = -\alpha m \frac{\beta}{t}. \tag{3.11}$$

By using a Chernoff bound [65] of the type $\Pr[X > \mathbb{E}[X] + u], \Pr[X < \mathbb{E}[X] - u] \leq e^{-2u^2/m}$ with $u > 0$ and setting $u = \pm\mathbb{E}[X] > 0$, we can make

$$\Pr[X > 0 | B_f(x,\sigma) = \overline{w}],\ \Pr[X < 0 | B_f(x,\sigma) = w] \leq \epsilon \tag{3.12}$$

by taking $m = O\big((\frac{t}{\alpha\beta})^2 \log \frac{1}{\epsilon}\big)$. Therefore Alice and Bob can decide if $B_f(x,\sigma) = w$ or $B_f(x,\sigma) = \overline{w}$ with error probability ϵ and $O\big((\frac{t}{\alpha\beta})^2 \log \frac{1}{\epsilon} \log n\big)$ bits of communication. ■

3.2.2 Quantum Upper Bound

Consider the f-BHP$_n^{\alpha,t}$ problem for $f : \{-1,1\}^t \to \{-1,1\}$ with $\mathrm{sdeg}(f) = 2$. Let $p : \{-1,1\}^t \to [-1,1]$ be a normalized sign-representing polynomial for f. Let $\beta = \min_x |p(x)|$ be again the bias of p. In the following, define $\widetilde{x} = (1, x_1, \ldots, x_t)$.

In order to obtain our upper bound, we borrow the idea of *block-multilinear* polynomials from [2, 3], which are also known as multilinear forms. We say that a polynomial q of degree k is block-multilinear if its variables $x_1, \ldots, x_N$ can be partitioned into k blocks $R_1, \ldots, R_k$, such that every monomial of q contains exactly one variable from each block. As a special case, a block-multilinear polynomial q of degree 2 can be written as

$$q(x_1, \ldots, x_n, y_1, \ldots, y_m) = \sum_{\substack{i\in[n] \\ j\in[m]}} a_{ij} x_i y_j \tag{3.13}$$

with variables in the first block labeled as $x_1, \ldots, x_n$ and the variables in the second block labeled as $y_1, \ldots, y_m$. Defining the matrix $A = (a_{ij})_{i\in[n],j\in[m]}$, then

$$q(x,y) = x^T A y \tag{3.14}$$

for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. We say that q is *bounded* if $|q(x,y)| \leq 1$ for all $x \in \{-1,1\}^n, y \in \{-1,1\}^m$. This translates to

$$\max_{\substack{x\in\{-1,1\}^n \\ y\in\{-1,1\}^m}} \left|\sum_{\substack{i\in[n] \\ j\in[m]}} a_{ij} x_i y_j\right| \leq 1, \tag{3.15}$$

i.e., $\|A\|_{\infty\to 1} \leq 1$. More generally, in the following, given a complex matrix M, we define $\|M\|_{p\to q} := \sup_{x\neq 0} \|Mx\|_q / \|x\|_p$ and $\|M\| := \|M\|_{2\to 2}$ is the spectral norm.

We shall also make use of the following results (a similar version of Theorem 3.3 was also proved in [159]).

Lemma 3.2 ([3, Lemma 7]). *Given an $m \times m$ complex matrix M, there is a unitary U (on a possibly larger space with basis $|1\rangle, \ldots, |k\rangle$ for some $k \geq m$) such that, for any unit vector $|y\rangle = \sum_{i=1}^{m} \alpha_i |i\rangle$, $U|y\rangle = \frac{M|y\rangle}{\|M\|} + |\phi\rangle$, where $|\phi\rangle$ consists of basis states $|i\rangle$, $i > m$ only.*

Theorem 3.3 ([3, Theorem 4]). *Let $p : \{-1,1\}^t \to [-1,1]$ be a sign-representing polynomial for f with $\mathrm{sdeg}(f) = 2$. Then there is a block-multilinear polynomial $\widetilde{p} : \mathbb{R}^{2(t+1)} \to \mathbb{R}$ such that $\widetilde{p}(\widetilde{x}, \widetilde{x}) = p(x)$ for any $x \in \{-1,1\}^t$, and $|\widetilde{p}(y)| \leq 3$ for any $y \in \{-1,1\}^{2(t+1)}$.*

Let $\widetilde{p} : \mathbb{R}^{2(t+1)} \to \mathbb{R}$ be the block-multilinear polynomial of degree 2 obtained from the sign-representing polynomial p of f according to Theorem 3.3. It can be written as

$$\widetilde{p}(x,y) = \sum_{i,j \in [t+1]} a_{ij} x_i y_j = x^T A y, \tag{3.16}$$

where $A = (a_{ij})_{i,j \in [t+1]}$.

With these in hands, we present our upper bound.

Theorem 3.4. $Q_\epsilon^1(f\text{-BHP}_n^{\alpha,t}) = O\left((\frac{t}{\alpha\beta})^2 \log \frac{1}{\epsilon} \log n\right)$ *if* $\mathrm{sdeg}(f) \leq 2$, *where β is the bias of any normalized sign-representing polynomial for f with degree ≤ 2.*

Proof. Consider the following protocol: Alice sends to Bob $m = O\big((\frac{t}{\alpha\beta})^2 \log \frac{1}{\epsilon}\big)$ copies of the quantum state of $O(\log n)$ qubits

$$|\psi_A\rangle = \frac{1}{\sqrt{n + n/t}} \left(\sum_{i=1}^{n} x_i |i\rangle + \sum_{i=1}^{n/t} |n+i\rangle \right). \tag{3.17}$$

Bob measures each of them by using the POVM

$$\left\{ |n+j\rangle\langle n+j| + \sum_{i=(j-1)t+1}^{jt} |\sigma^{-1}(i)\rangle\langle\sigma^{-1}(i)| \right\}_{j \in [n/t]}, \tag{3.18}$$

where $\sigma \in \mathbb{S}_n$ is his permutation, and attaches a qubit in the state $|+\rangle$ to each of the resulting states. Let $I \subseteq [n/t]$ be the sequence of indices from his measurements. Then his state is

$$|\psi_B\rangle = \bigotimes_{j \in I} |+\rangle |\psi^{(j)}\rangle, \tag{3.19}$$

where

$$|\psi^{(j)}\rangle = \frac{1}{\sqrt{t+1}} \left(|n+j\rangle + \sum_{i=(j-1)t+1}^{jt} x_{\sigma^{-1}(i)} |\sigma^{-1}(i)\rangle \right). \tag{3.20}$$

Let A be the $(t+1) \times (t+1)$ matrix from the representation of $\widetilde{p}$ according to Eq. (3.16). Lemma 3.2 guarantees the existence of a unitary U_j such that $U_j|\psi^{(j)}\rangle = \frac{A|\psi^{(j)}\rangle}{\|A\|} + |\phi^{(j)}\rangle$, with $\langle\phi^{(j)}|\psi^{(j)}\rangle = 0$. Bob then applies a controlled U_j gate onto each $|+\rangle_j |\psi^{(j)}\rangle$ to obtain

$$\bigotimes_{j \in I} CU_j |\psi_B\rangle = \bigotimes_{j \in I} \left(\frac{1}{\sqrt{2}} |0\rangle|\psi^{(j)}\rangle + \frac{1}{\sqrt{2}} |1\rangle U_j |\psi^{(j)}\rangle \right) \tag{3.21}$$

and then performs a Hadamard gate on the first qubit of each of the subsystems I and measures them. Let $m_j \in \{0,1\}$ be the result of the measurement for block $j \in I$. Define the random variable

$$X(j) := \begin{cases} (-1)^{m_j} w_j & \text{if } j \in [\alpha n/t], \\ 0 & \text{if } j \notin [\alpha n/t], \end{cases} \tag{3.22}$$

and define $X := \sum_{j \in I} X(j)$. Bob then computes $\mathrm{sgn}(X)$: if $\mathrm{sgn}(X) > 0$, he outputs that $B_f(x,\sigma) = w$, and if $\mathrm{sgn}(X) < 0$, he outputs that $B_f(x,\sigma) = \overline{w}$.

To see why the protocol works, first note that the probability of measuring 0 is

$$\Pr[0] = \frac{1}{2}\left(1 + \langle \psi^{(j)} | U | \psi^{(j)} \rangle\right) = \frac{1}{2}\left(1 + \frac{\langle \psi^{(j)} | A | \psi^{(j)} \rangle}{\|A\|}\right) \tag{3.23}$$

$$= \frac{1}{2}\left(1 + \frac{\widetilde{p(\sigma(x)^{(j)}}, \widetilde{\sigma(x)^{(j)})}}{\|A\|(t+1)}\right) = \frac{1}{2}\left(1 + \frac{p(\sigma(x)^{(j)})}{\|A\|(t+1)}\right). \tag{3.24}$$

The remainder of the argument is similar to Theorem 3.1. Recalling that $m = |I|$, the expected value of X is

$$\mathbb{E}[X] = m \cdot \mathbb{E}_j[X(j)] \tag{3.25}$$

$$= \alpha m \cdot \mathbb{E}_j[(-1)^{m_j} w_j] \tag{3.26}$$

$$= \alpha m \frac{t}{n} \sum_{j=1}^{n/t} \left(\Pr[m_j = 0] - \Pr[m_j = 1]\right) w_j \tag{3.27}$$

$$= \alpha m \frac{t}{n} \left[\sum_{j : w_j = 1} \frac{p(\sigma(x)^{(j)})}{\|A\|(t+1)} - \sum_{j : w_j = -1} \frac{p(\sigma(x)^{(j)})}{\|A\|(t+1)} \right]. \tag{3.28}$$

If $f(\sigma(x)^{(j)}) = w_j$, then $w_j = 1 \implies p(\sigma(x)^{(j)}) \geq \beta > 0$ and $w_j = -1 \implies p(\sigma(x)^{(j)}) \leq -\beta < 0$. Therefore

$$\mathbb{E}[X] \geq \alpha m \frac{t}{n} \frac{1}{\|A\|(t+1)} \left[\sum_{j : w_j = 1} \beta - \sum_{j : w_j = -1} -\beta \right] = \frac{\alpha m \beta}{\|A\|(t+1)}. \tag{3.29}$$

If, on the other hand, $f(\sigma(x)^{(j)}) = -w_j$, then $w_j = 1 \implies p(\sigma(x)^{(j)}) \leq -\beta < 0$ and $w_j = -1 \implies p(\sigma(x)^{(j)}) \geq \beta > 0$. Therefore

$$\mathbb{E}[X] \leq \alpha m \frac{t}{n} \frac{1}{\|A\|(t+1)} \left[\sum_{j : w_j = 1} -\beta - \sum_{j : w_j = -1} \beta \right] = -\frac{\alpha m \beta}{\|A\|(t+1)}. \tag{3.30}$$

By using a Chernoff bound [65] of the type $\Pr[X > \mathbb{E}[X] + u], \Pr[X < \mathbb{E}[X] - u] \leq e^{-2u^2/m}$ with $u > 0$ and setting $u = \pm \mathbb{E}[X] > 0$, we can make

$$\Pr[X > 0 | B_f(x,\sigma) = \overline{w}], \ \Pr[X < 0 | B_f(x,\sigma) = w] \leq \epsilon \tag{3.31}$$

by taking $m = O\big((\frac{t}{\alpha\beta})^2 \log \frac{1}{\epsilon}\big)$, where we use that $\|A\| \leq \|A\|_{\infty\to 1} \leq 3$ according to Theorem 3.3 (note that $\frac{\|Ax\|_2}{\|x\|_2} \leq \frac{\|Ax\|_1}{\|x\|_\infty}$, and taking supremum over all x on both sides gives $\|A\| \leq \|A\|_{\infty\to 1}$). Therefore Alice and Bob can decide if $B_f(x,\sigma) = w$ or $B_f(x,\sigma) = \overline{w}$ with error probability ϵ and $O\big((\frac{t}{\alpha\beta})^2 \log \frac{1}{\epsilon} \log n\big)$ qubits of communication. ■

3.3 Reductions from the Boolean Hidden Matching problem

As mentioned before, in [81] it was proved that the Boolean Hidden Partition problem using PARITY on 2 bits (aka the BHM problem) is hard to solve, i.e., $R^1(\text{BHM}) = \Omega(\sqrt{n/\alpha})$. With this result alone it is possible to prove that the f-Boolean Hidden Partition problem for almost any symmetric Boolean function with $\text{sdeg}(f) \geq 2$ is at least as hard to solve. This can be achieved via a simple reduction from the BHM problem to the f-BHP$_n^{\alpha,t}$ problem with symmetric functions, which we shall show in this section.

For this section, in a slight abuse of notation we define $|x| = |\{i : x_i = -1\}|$ to be the "Hamming weight" of x. Let $s, t \in \mathbb{N}$, with $s \leq t$. Consider a symmetric Boolean function $f_s : \{-1,1\}^t \to \{-1,1\}$ such that (without loss of generality) $f_s(1^n) = 1$ and

$$f_s(x) = \begin{cases} +1 \text{ if } 0 \leq |x| \leq \theta_1 \text{ or } & \theta_{2i} < |x| \leq \theta_{2i+1}, i = 1, 2 \ldots, \lfloor s/2 \rfloor, \\ -1 \text{ if } & \theta_{2j-1} < |x| \leq \theta_{2j}, j = 1, 2, \ldots, \lfloor (s+1)/2 \rfloor, \end{cases} \tag{3.32}$$

where $\theta_k \in \mathbb{N}$ for $k = 1, \ldots, s+1$ and $0 \leq \theta_1 < \cdots < \theta_s < \theta_{s+1} = t$ and $\theta_{k+1} - \theta_k \geq 1$ for all $k = 1, \ldots, s$. The following result from [15] tells us that $\text{sdeg}(f_s) = s$.

Lemma 3.5 ([15, Lemma 2.6])**.** *If f is a symmetric function, then* sdeg(f) *is equal to the number of times f changes sign when expressed as a univariate function in $\sum_i x_i$.*

In order to reduce f_s-BHP$_n^{\alpha,t}$ from BHM we first need to reduce the function f_s from PARITY, i.e., we want that $\forall x' \in \{-1,1\}^2$, $\exists x \in \{-1,1\}^t$ such that $f_s(x) = \text{PARITY}(x')$. The key combinatorial step to achieve this is shown in the next Lemma.

Lemma 3.6. *Let $f_s : \{-1,1\}^t \to \{-1,1\}$ be the symmetric Boolean function from Eq. (3.32) with $s \geq 2$ such that either $2|t$ or $\theta_2 - \theta_1 < t-1$. Then there are $a, b \in \mathbb{N}$ such that $\forall x' \in \{-1,1\}^2$, $\exists x \in \{-1,1\}^t$ such that $f_s(x) = \text{PARITY}(x')$ and $|x| = a|x'| + b$.*

Proof. The condition that $\forall x' \in \{-1,1\}^2$, $\exists x \in \{-1,1\}^t$ such that $f_s(x) = \text{PARITY}(x')$ and $|x| = a|x'| + b$ is equivalent to

$$\begin{cases} |x'| = 0 \implies f_s(b) = 1, \\ |x'| = 1 \implies f_s(a+b) = -1, \\ |x'| = 2 \implies f_s(2a+b) = 1. \end{cases} \tag{3.33}$$

We divide the proof into two cases: either there is $k^* \in \{1, \dots, s-1\}$ such that $\theta_{k^*+1} - \theta_{k^*}$ is odd or there is not such k^*. Suppose first that such k^* exists. Without loss of generality we assume that $f_s(x) = -1$ for $\theta_{k^*} < |x| \leq \theta_{k^*+1}$, otherwise we just flip the values of f_s. Set

$$\begin{cases} a = (\theta_{k^*+1} - \theta_{k^*} + 1)/2, \\ b = \theta_{k^*}. \end{cases} \tag{3.34}$$

First, $a, b \in \mathbb{N}$. Second, $a + b = (\theta_{k^*+1} + \theta_{k^*} + 1)/2$, hence $\theta_{k^*} < a + b \leq \theta_{k^*+1}$, since $\theta_{k^*+1} - \theta_{k^*} \geq 1$. And third, $2a + b = \theta_{k^*+1} + 1 \leq \theta_{k^*+2}$. Thus all conditions (3.33) are satisfied.

Now suppose that $2|(\theta_{k+1} - \theta_k)$ for all $k = 1, \dots, s-1$. Define the bit $\delta = \mathbf{1}[\theta_1 \neq 0]$ and set

$$\begin{cases} a = (\theta_2 - \theta_1 + 2)/2, \\ b = \theta_1 - \delta. \end{cases} \tag{3.35}$$

First, $a, b \in \mathbb{N}$ (note that $\delta = 1 \implies \theta_1 > 0$). Second, $a + b = (\theta_2 + \theta_1 + 2 - 2\delta)/2$, hence $\theta_1 < a + b \leq \theta_2$, since $\theta_2 - \theta_1 \geq 2$ by hypothesis. And third, $2a + b = \theta_2 + 2 - \delta \leq t$ since $\theta_2 - \theta_1 \leq t - 1$ and $\theta_2 < t$ (so $\theta_2 = t - 1 \implies \delta = 1$). Thus all conditions (3.33) are satisfied. ■

If $2 \nmid t$ and $\theta_2 - \theta_1 = t - 1$, then our conditions give us

$$\begin{cases} b = 0, \\ 0 < a < t, \\ 2a = t, \end{cases} \tag{3.36}$$

and we see that the condition $2a = t$ cannot be fulfilled by $a \in \mathbb{N}$. This case corresponds to the symmetric Boolean function Not All Equal (NAE), defined by $\mathrm{NAE}(x) = 1$ if $|x| \in \{0, t\}$ and $\mathrm{NAE}(x) = -1$ otherwise, with t odd.

Given the reduction above from PARITY to f_s, we can construct our reduction from the BHM problem to the f_s-$\mathrm{BHP}_n^{\alpha,t}$ problem. In the following theorem, we shall write $y^{(j;t)} = (y_{(j-1)t+1}, y_{(j-1)t+2}, \dots, y_{jt}) \in \{-1, 1\}^t$ to stress the size t of the blocks from y in order to better differentiate between strings in the reduction.

Theorem 3.7. *Let $f_s : \{-1, 1\}^t \to \{-1, 1\}$ be the symmetric Boolean function from Eq. (3.32) with $s \geq 2$ such that either $2|t$ or $\theta_2 - \theta_1 < t - 1$. Then $R^1(f_s\text{-}\mathrm{BHP}_n^{\alpha,t}) = \Omega(\sqrt{n/(\alpha t)})$.*

Proof. Suppose towards a contradiction that $R^1(f_s\text{-}\mathrm{BHP}_n^{\alpha,t}) = o(\sqrt{n/(\alpha t)})$, i.e., there is a protocol $\mathcal{P}$ that solves f_s-$\mathrm{BHP}_n^{\alpha,t}$ with $o(\sqrt{n/(\alpha t)})$ bits of communication. We are going to show that such protocol would allow Alice and Bob to solve the BHM problem with $o(\sqrt{n/\alpha})$ bits of communication, which leads to a contradiction.

Let $a, b \in \mathbb{N}$ be the numbers used in reducing f_s from PARITY in Lemma 3.6. Alice increases her bit string $x \in \{-1, 1\}^n$ as follows: she makes a copies of x, obtaining $x^a \in \{-1, 1\}^{an}$, where

$x^a = xx\cdots x$ represents x repeated a times. She then adds $bn/2$ times the bit 1, obtaining $x^a 1^{bn/2}$. Finally, she adds $(t-2a-b)n/2$ times the bit -1, to finally obtain $x_f = x^a 1^{bn/2}\text{-}1^{(t-2a-b)n/2}$. Note that $x_f \in \{-1,1\}^{nt/2}$.

Bob, on the other hand, increases his permutation $\sigma \in \mathbb{S}_n$ to a new permutation $\sigma_f \in \mathbb{S}_{nt/2}$. In order to describe how he does this, we ease the notation by referring to the j-th block $(\pi^{-1}((j-1)t+1),\ldots,\pi^{-1}(jt))$ of a given permutation π as $(B_{j,1},\ldots,B_{j,t})$. With this notation, the j-th block $(B_{j,1},B_{j,2})$ of the permutation σ is mapped to the j-th block

$$\Big(B_{j,1},\ B_{j,2},\ n+B_{j,1},\ n+B_{j,2},\ \ldots,\ (a-1)n+B_{j,1},\ (a-1)n+B_{j,2},\\ an+j,\ an+j+\frac{n}{2},\ \ldots,\ an+j+(t-2a-1)\frac{n}{2}\Big) \tag{3.37}$$

of the new permutation σ_f. Note that the new block has t elements, as expected.

Consider the strings $\sigma_f(x_f)^{(j;t)} \in \{-1,1\}^t$ and $\sigma(x)^{(j;2)} \in \{-1,1\}^2$, with $j = 1,\ldots,n/2$. By construction we have that $|\sigma_f(x_f)^{(j;t)}| = a|\sigma(x)^{(j;2)}| + b$ and, according to Lemma 3.6, we get $f_s(\sigma_f(x_f)^{(j;t)}) = \mathrm{PARITY}(\sigma(x)^{(j;2)})$ for all $j = 1,\ldots,n/2$. Hence we see that every instance of the problem $\mathrm{BHM} : \{-1,1\}^n \to \{-1,1\}$ is mapped to an instance of the problem $f_s\text{-BHP}_n^{\alpha,t} : \{-1,1\}^{nt/2} \to \{-1,1\}$. Therefore we could map the BHM problem into the $f_s\text{-BHP}_n^{\alpha,t}$ problem and use the protocol $\mathscr{P}$ in order to solve it with $o(\sqrt{n/(\alpha t)})$ bits of communication, which is impossible. Thus $R^1(f_s\text{-BHP}_n^{\alpha,t}) = \Omega(\sqrt{n/(\alpha t)})$. ■

3.4 Classical Lower Bound

Given a Boolean function $f : \{-1,1\}^t \to \{-1,1\}$, in this section we shall prove that the associated f-Boolean Hidden Partition problem is hard if $\mathrm{phdeg}(f) \geq 2$. The proof follows the general idea of [81, 192], but, by using ideas borrowed from [177], the technical execution was substantially changed, as we now derive the lower bound through an upper bound on the bias ϵ_{bias} of the protocol, i.e., its success probability minus a half. As a consequence, Parseval's identity is not central any more and the bound comes from a single application of the KKL inequality.

By Yao's minimax principle [205], it suffices to analyse *deterministic* protocols under a suitable "hard" input distribution. We choose Alice's input x and Bob's input σ independently and uniformly over $\{-1,1\}^n$ and $\mathbb{S}_n$, respectively. The input distribution is completed by choosing $w = B_f(x,\sigma)$ with probability $1/2$ and $w = \overline{B_f(x,\sigma)}$ with probability $1/2$. Recall that the total variation distance[2] between two distributions D and D' is defined as $\|D-D'\|_{\mathrm{tvd}} := \sum_x |D(x)-D'(x)|$.

[2]This distance is often defined with a factor of $1/2$; here we use the same normalisation as [81].

We briefly mention that the upper bound below on α comes from technical reasons that will arise during the proof. It is not tight, though, and could possibly be improved up to $\alpha \leq 1/2$.

Theorem 3.8. *If* $\mathrm{phdeg}(f) = d \geq 1$ *and* $\alpha \leq (2\hat{\|}f\hat{\|}_1^2)^{-1/d}$, *where* $\hat{\|}f\hat{\|}_1 := \sum_{T\subseteq[t]} |\widehat{f}(T)|$, *then for sufficiently small constant* $\epsilon > 0$, $R_\epsilon^1(f\text{-BHP}_n^{\alpha,t}) = \Omega((\alpha n/t)^{1-1/d})$.

Proof. For $d = 1$ the theorem is trivial, so assume $d \geq 2$. Let $\epsilon > 0$ be a sufficiently small constant. Consider a classical deterministic protocol that sends one message of length at most C bits. The message can be thought of as partitioning the set of 2^n x's into 2^C disjoint sets $A_1, \ldots, A_{2^C}$, one for each message. These sets have size 2^{n-C} on average. Also, by a counting argument, at most a $2^{-\ell}$-fraction of all $x \in \{-1,1\}^n$ can sit in sets of size $\leq 2^{n-C-\ell}$. Therefore, taking $\ell = \log(1/\epsilon)$, with probability at least $1-\epsilon$, the message of Alice corresponds to a set $A \subseteq \{-1,1\}^n$ of size at least $2^{n-C-\log(1/\epsilon)}$. Set $c := C + \log(1/\epsilon)$ such that this size is 2^{n-c}.

Let X be uniformly distributed over $A \in \{-1,1\}^n$ such that $|A| \geq 2^{n-c}$ and let $Z = B_f(X,\sigma)$ given Bob's permutation σ. Bob, by looking at w, needs to decide whether $w = Z$ or $w = \overline{Z}$, i.e., he needs to discriminate between the following two induced distributions,

$$p_\sigma(z) = \frac{|\{x \in A | B_f(x,\sigma) = z\}|}{|A|} \text{ and } q_\sigma(z) = \frac{|\{x \in A | B_f(x,\sigma) = \overline{z}\}|}{|A|}. \tag{3.38}$$

He can only achieve this if the distributions have large total variation distance, since it is well known that the best success probability for distinguishing two distributions q_1 and q_2 with one sample is $1/2 + \|q_1 - q_2\|_{\mathrm{tvd}}/4$. We are then going to upper bound the expectation of the bias over Bob's permutation and, by demanding a small distributional error, we shall obtain a lower bound $c = \Omega(n^{1-1/d})$.

We start upper-bounding the total variation distance by using Jensen's inequality,

$$\mathop{\mathbb{E}}_\sigma[\|p_\sigma - q_\sigma\|_{\mathrm{tvd}}] \leq \sqrt{\mathop{\mathbb{E}}_\sigma[\|p_\sigma - q_\sigma\|_{\mathrm{tvd}}^2]}, \tag{3.39}$$

and then by the Cauchy-Schwarz inequality we obtain

$$\mathop{\mathbb{E}}_\sigma[\|p_\sigma - q_\sigma\|_{\mathrm{tvd}}^2] \leq 2^{2\alpha n/t} \mathop{\mathbb{E}}_\sigma[\|p_\sigma - q_\sigma\|_2^2]. \tag{3.40}$$

By using Parseval's identity (Lemma 1.22) we finally get

$$\mathop{\mathbb{E}}_\sigma[\|p_\sigma - q_\sigma\|_2^2] = \mathop{\mathbb{E}}_\sigma\left[\sum_{\substack{V\subseteq[\alpha n/t]\\ V\neq\emptyset}} \widehat{r_\sigma}(V)^2\right], \tag{3.41}$$

where $r_\sigma(z) = p_\sigma(z) - q_\sigma(z)$, and we observe that $\widehat{p_\sigma}(\emptyset) = \widehat{q_\sigma}(\emptyset)$, as p_σ and q_σ are probability distributions. This means that the expected bias ϵ_{bias} satisfies

$$\epsilon_{bias} \leq \frac{1}{4} 2^{2\alpha n/t} \mathop{\mathbb{E}}_\sigma\left[\sum_{\substack{V\subseteq[\alpha n/t]\\ V\neq\emptyset}} \widehat{r_\sigma}(V)^2\right]. \tag{3.42}$$

We can show that p_σ and q_σ are close for most permutations σ by bounding the Fourier coefficients of r_σ, which are related to the Fourier coefficients of g as follows.

$$\widehat{r_\sigma}(V) = \frac{1}{2^{\alpha n/t}} \sum_{z\in\{-1,1\}^{\alpha n/t}} (p_\sigma(z) - q_\sigma(z))\,\chi_V(z) \tag{3.43}$$

$$= \frac{1}{|A|2^{\alpha n/t}} \sum_{\substack{z\in\{-1,1\}^{\alpha n/t}\\ x\in\{-1,1\}^n}} \mathbf{1}[x\in A]\,(\mathbf{1}[B_f(x,\sigma)=z] - \mathbf{1}[B_f(x,\sigma)=\overline{z}])\,\chi_V(z) \tag{3.44}$$

$$= \frac{2}{|A|2^{\alpha n/t}} \sum_{x\in\{-1,1\}^n} g(x)\chi_V(B_f(x,\sigma)) \tag{3.45}$$

for $|V|$ odd, otherwise $\widehat{r_\sigma}(V)=0$, and where $g:\{-1,1\}^n\to\{0,1\}$ is the characteristic function of A, i.e., $g(x)=1$ iff $x\in A$. Using the Fourier expansion of f and setting $|V|=k$, we have

$$\chi_V(B_f(x,\sigma)) = \prod_{j\in V}\left(\sum_{T\subseteq[t]} \widehat{f}(T)\chi_T(\sigma(x)^{(j)})\right) \tag{3.46}$$

$$= \prod_{j=1}^{k}\left(\sum_{T_j\subseteq[t]} \widehat{f}(T_j)\chi_{T_j}(\sigma(x)^{(V_j)})\right) \tag{3.47}$$

$$= \sum_{T_1,\ldots,T_k\subseteq[t]} \widehat{f}(T_1)\cdots\widehat{f}(T_k)\chi_{T_1}(\sigma(x)^{(V_1)})\cdots\chi_{T_k}(\sigma(x)^{(V_k)}) \tag{3.48}$$

$$= \sum_{T_1,\ldots,T_k\subseteq[t]} \widehat{f}(T_1)\cdots\widehat{f}(T_k)\chi_{V_1[T_1]\cup V_2[T_2]\cup\cdots\cup V_k[T_k]}(\sigma(x)) \tag{3.49}$$

$$= \sum_{T_1,\ldots,T_k\subseteq[t]} \widehat{f}(T_1)\cdots\widehat{f}(T_k)\chi_{V\bullet T}(\sigma(x)), \tag{3.50}$$

where at the end we use the notation $V_i[T_i]$ to denote subset T_i being positioned in block V_i, and then use the notation $V\bullet T$ to compactly represent $V_1[T_1]\cup V_2[T_2]\cup\cdots\cup V_k[T_k]$. So we have

$$\widehat{r_\sigma}(V) = \frac{2}{|A|2^{\alpha n/t}} \sum_{\substack{x\in\{-1,1\}^n\\ T_1,\ldots,T_k\subseteq[t]}} g(x)\widehat{f}(T_1)\cdots\widehat{f}(T_k)\chi_{V\bullet T}(\sigma(x)) \tag{3.51}$$

$$= \frac{2^{n+1}}{|A|2^{\alpha n/t}} \sum_{T_1,\ldots,T_k\subseteq[t]} \widehat{f}(T_1)\cdots\widehat{f}(T_k)\widehat{g}(\sigma^{-1}(V\bullet T)), \tag{3.52}$$

using that, for all $U\subseteq[n]$,

$$\chi_U(\sigma(x)) = \prod_{i\in U}\sigma(x)_i = \prod_{i\in U} x_{\sigma^{-1}(i)} = \prod_{\sigma(j)\in U} x_j = \prod_{j\in\sigma^{-1}(U)} x_j. \tag{3.53}$$

With all that, Eq. (3.42) becomes

$$\epsilon_{bias} \le \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1\\ k\text{ odd}}}^{\alpha n/t} \sum_{\substack{V\subseteq[\alpha n/t]\\ |V|=k}} \mathop{\mathbb{E}}_{\sigma}\left[\left(\sum_{T_1,\ldots,T_k\subseteq[t]} \widehat{f}(T_1)\cdots\widehat{f}(T_k)\cdot\widehat{g}(\sigma^{-1}(V\bullet T))\right)^2\right] \tag{3.54}$$

$$= \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{V \subseteq [\alpha n/t] \\ |V|=k}} \sum_{\substack{T_1,\ldots,T_k \subseteq [t] \\ U_1,\ldots,U_k \subseteq [t]}} \mathop{\mathbb{E}}_{\sigma} \left[\widehat{g}(\sigma^{-1}(V \bullet T)) \widehat{g}(\sigma^{-1}(V \bullet U)) \right] \prod_{j=1}^{k} \widehat{f}(T_j) \widehat{f}(U_j) \tag{3.55}$$

$$\leq \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{V \subseteq [\alpha n/t] \\ |V|=k}} \left(\sum_{T_1,\ldots,T_k \subseteq [t]} \sqrt{\mathop{\mathbb{E}}_{\sigma} \left[\widehat{g}(\sigma^{-1}(V \bullet T))^2 \right]} \prod_{j=1}^{k} |\widehat{f}(T_j)| \right)^2, \tag{3.56}$$

where we used that $\mathbb{E}[XY] \leq \sqrt{\mathbb{E}[X^2]\mathbb{E}[Y^2]}$. Now we use the following combinatorial Lemma.

Lemma 3.9. *For all $S \subseteq [n]$ and all $T_1, \ldots, T_k \subseteq [t]$ such that $|S| = \sum_{j=1}^{k} |T_j|$,*

$$\mathop{\mathbb{E}}_{\sigma \in \mathbb{S}_n} \left[\widehat{g}(\sigma^{-1}(V \bullet T))^2 \right] = \frac{1}{\binom{n}{|S|}} \sum_{\substack{S \subseteq [n] \\ |S| = \sum_{j=1}^{k} |T_j|}} \widehat{g}(S)^2. \tag{3.57}$$

Proof. By the definition of expected value,

$$\mathop{\mathbb{E}}_{\sigma \in \mathbb{S}_n} \left[\widehat{g}(\sigma^{-1}(V \bullet T))^2 \right] = \frac{1}{n!} \sum_{\sigma \in \mathbb{S}_n} \widehat{g}(\sigma^{-1}(V \bullet T))^2 \tag{3.58}$$

$$= \frac{1}{n!} \sum_{\sigma \in \mathbb{S}_n} \sum_{\substack{S \subseteq [n] \\ |S| = \sum_{j=1}^{k} |T_j|}} \mathbf{1}[\sigma^{-1}(V \bullet T) = S] \cdot \widehat{g}(S)^2 \tag{3.59}$$

$$= \frac{1}{\binom{n}{|S|}} \sum_{\substack{S \subseteq [n] \\ |S| = \sum_{j=1}^{k} |T_j|}} \widehat{g}(S)^2, \tag{3.60}$$

since $\sum_{\sigma \in \mathbb{S}_n} \mathbf{1}[\sigma^{-1}(V \bullet T) = S] = |\{\sigma \in \mathbb{S}_n \mid \sigma^{-1}(V \bullet T) = S\}| = |S|!(n - |S|)!$. ∎

Using this Lemma in Eq. (3.56), we have

$$\epsilon_{bias} \leq \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{V \subseteq [\alpha n/t] \\ |V|=k}} \left(\sum_{T_1,\ldots,T_k \subseteq [t]} \sqrt{\sum_{\substack{S \subseteq [n] \\ |S| = \sum_{i=1}^{k} |T_i|}} \frac{\widehat{g}(S)^2}{\binom{n}{|S|}}} \prod_{j=1}^{k} |\widehat{f}(T_j)| \right)^2 \tag{3.61}$$

$$\leq \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{V \subseteq [\alpha n/t] \\ |V|=k}} \left(\sqrt{\sum_{\substack{S \subseteq [n] \\ kd \leq |S| \leq kt}} \frac{\widehat{g}(S)^2}{\binom{n}{|S|}}} \sum_{T_1,\ldots,T_k \subseteq [t]} \prod_{j=1}^{k} |\widehat{f}(T_j)| \right)^2 \tag{3.62}$$

$$= \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{S \subseteq [n] \\ kd \leq |S| \leq kt}} \frac{\binom{\alpha n/t}{k}}{\binom{n}{|S|}} \cdot \widehat{g}(S)^2 \left(\sum_{T_1,\ldots,T_k \subseteq [t]} \prod_{j=1}^{k} |\widehat{f}(T_j)| \right)^2 \tag{3.63}$$

$$= \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{S\subseteq[n] \\ kd\le|S|\le kt}} \hat{\|}f\hat{\|}_1^{2k} \frac{\binom{\alpha n/t}{k}}{\binom{n}{|S|}} \cdot \widehat{g}(S)^2, \tag{3.64}$$

where Eq. (3.62) comes from expanding the constraint $|S| = \sum_{j=1}^{k} |T_j|$ to the interval $kd \le |S| \le kt$, since $d \le |T_j| \le t$ for all $j \in [k]$ ($\widehat{f}(T_j) = 0$ if $|T_j| < \mathrm{phdeg}(f) = d$), so that the summation on $S \subseteq [n]$ can be pulled out of the summation on $T_1, \dots, T_k \subseteq [t]$, and in Eq. (3.64) we denoted the sum of the Fourier masses of f by $\hat{\|}f\hat{\|}_1 := \sum_{T\subseteq[t]} |\widehat{f}(T)|$.

By using the upper bounds $\binom{an}{m} \le a^m \binom{n}{m}$ for $a \le 1$ in Eq. (3.64) to get Eq. (3.65) below and then $\binom{n}{m}\binom{ln}{lm}^{-1} \le (\frac{m}{n})^{(l-1)m}$ (see [177, Appendix A.5]) to get Eq. (3.66) below, we obtain

$$\epsilon_{bias} \le \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{S\subseteq[n] \\ kd\le|S|\le kt}} \hat{\|}f\hat{\|}_1^{2k} \left(\frac{\alpha|S|}{kt}\right)^k \frac{\binom{nk/|S|}{k}}{\binom{n}{|S|}} \cdot \widehat{g}(S)^2 \tag{3.65}$$

$$\le \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{S\subseteq[n] \\ kd\le|S|\le kt}} \hat{\|}f\hat{\|}_1^{2k} \alpha^k \left(\frac{|S|}{n}\right)^{|S|-k} \widehat{g}(S)^2 \tag{3.66}$$

$$\le \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{S\subseteq[n] \\ kd\le|S|\le kt}} \hat{\|}f\hat{\|}_1^{2k} \alpha^{|S|} \left(\frac{kt}{\alpha n}\right)^{|S|-k} \widehat{g}(S)^2 \tag{3.67}$$

$$\le \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{S\subseteq[n] \\ kd\le|S|\le kt}} \hat{\|}f\hat{\|}_1^{2k} \alpha^{kd} \left(\left(\frac{kt}{\alpha n}\right)^{1-1/d}\right)^{|S|} \widehat{g}(S)^2 \tag{3.68}$$

$$= \frac{2^{2n}}{|A|^2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \left(\alpha^d \hat{\|}f\hat{\|}_1^2\right)^k \sum_{\substack{S\subseteq[n] \\ kd\le|S|\le kt}} \delta_k^{|S|} \widehat{g}(S)^2, \tag{3.69}$$

where we defined $\delta_k = (kt/\alpha n)^{1-1/d} \le 1$. We now apply the KKL inequality (Lemma 1.29) as

$$\frac{2^{2n}}{|A|^2} \sum_{\substack{S\subseteq[n] \\ kd\le|S|\le kt}} \delta_k^{|S|} \widehat{g}(S)^2 \le \frac{2^{2n}}{|A|^2} \sum_{S\subseteq[n]} \delta_k^{|S|} \widehat{g}(S)^2 \le \left(\frac{2^n}{|A|}\right)^{2\delta_k/(1+\delta_k)} \le 2^{\delta_k c}, \tag{3.70}$$

using that $|A| \ge 2^{n-c}$. With this and naming $\widetilde{\alpha} = \alpha^d \hat{\|}f\hat{\|}_1^2$, so that $\alpha \le (2\hat{\|}f\hat{\|}_1^2)^{-1/d} \implies \widetilde{\alpha} \le 1/2$, we finally get

$$\epsilon_{bias} \le \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \widetilde{\alpha}^k 2^{\delta_k c} = \widetilde{\alpha} 2^{\delta_1 c} + \sum_{\substack{k=3 \\ k \text{ odd}}}^{\alpha n/t} \widetilde{\alpha}^{k/2} \widetilde{\alpha}^{k/2} 2^{\delta_k c} \tag{3.71}$$

$$\le \widetilde{\alpha} 2^{\delta_1 c} + \left(\sum_{\substack{k=3 \\ k \text{ odd}}}^{\alpha n/t} \widetilde{\alpha}^{k/2}\right) \max_{\substack{3\le k\le \alpha n/t \\ k \text{ odd}}} \widetilde{\alpha}^{k/2} 2^{\delta_k c} \le \widetilde{\alpha} 2^{\delta_1 c} + \frac{\widetilde{\alpha}^{3/2}}{1-\widetilde{\alpha}} \max_{\substack{3\le k\le \alpha n/t \\ k \text{ odd}}} \widetilde{\alpha}^{k/2} 2^{\delta_k c}. \tag{3.72}$$

For sufficiently small distributional error (such that $2\epsilon_{bias} \geq 0.95$) and for $0 \leq \widetilde{\alpha} \leq 0.5$, we have either $\frac{\widetilde{\alpha}^{3/2}}{1-\widetilde{\alpha}}\widetilde{\alpha}^{k/2}2^{\delta_k c} \geq \widetilde{\alpha}^{3/2}$ for some $k \geq 3$ or $\widetilde{\alpha}2^{\delta_1 c} \geq 0.95 - \widetilde{\alpha}^{3/2} \geq \widetilde{\alpha}^{0.9}$. In the first case, we have $2^{\delta_k c} \geq (1-\widetilde{\alpha})\widetilde{\alpha}^{-k/2}$ and so

$$c = \Omega\left(\frac{k}{\delta_k}\log\frac{1}{\widetilde{\alpha}}\right) = \Omega\left(\left(\frac{\alpha}{t}\right)^{1-1/d} k^{1/d}\log\left(\frac{1}{\alpha^d\hat{\|}f\hat{\|}_1}\right)n^{1-1/d}\right) \tag{3.73}$$

for $k \geq 3$. In the second case, $2^{\delta_1 c} \geq \widetilde{\alpha}^{-0.1}$ and thus

$$c = \Omega\left(\frac{1}{\delta_1}\log\frac{1}{\widetilde{\alpha}}\right) = \Omega\left(\left(\frac{\alpha}{t}\right)^{1-1/d}\log\left(\frac{1}{\alpha^d\hat{\|}f\hat{\|}_1}\right)n^{1-1/d}\right). \tag{3.74}$$

This concludes the proof. ■

3.5 Quantum Lower Bound

While in the previous section we proved a classical lower bound $\Omega(n^{1-1/d})$ for the f-BHP$_n^{\alpha,t}$ when $\mathrm{phdeg}(f) = d \geq 2$, in this section we shall prove its quantum analogue, Theorem 3.11. The proof follows the same line as the quantum lower bound for the BHH$_n^t$ problem from [177]. The 'hard' input distribution is still uniform on Alice's input $x \in \{-1,1\}^n$, Bob's input $\sigma \in \mathbb{S}_n$ and the function value $b \in \{-1,1\}$, which fixes Bob's second input $w = B_f(x,\sigma) \circ b^{\alpha n/t}$. Differently from the classical lower bound proof, the Fourier analysis is performed directly on the encoding messages and not on the pre-images of a fixed encoding message, since there is no clear quantum analogue of conditioning on a message. Moreover, the matrix-valued hypercontractive inequality is now used.

Lemma 3.10 ([98]). *Let ρ_0, ρ_1 be two quantum states which appear with probability p and $1-p$, respectively. The optimal success probability of predicting which state it is by a POVM is*

$$\frac{1}{2} + \frac{1}{2}\|p\rho_0 - (1-p)\rho_1\|_{\mathrm{tr}}. \tag{3.75}$$

Theorem 3.11. *If $\mathrm{phdeg}(f) = d \geq 3$ and $\alpha \leq (2\hat{\|}f\hat{\|}_1)^{-2/d}$, where $\hat{\|}f\hat{\|}_1 := \sum_{T\subseteq[t]}|\widehat{f}(T)|$, then for sufficiently small constant $\epsilon > 0$, $Q_\epsilon^1(f\text{-BHP}_n^{\alpha,t}) = \Omega((\alpha n/t)^{1-2/d})$.*

Proof. We fix an arbitrary quantum protocol $\mathscr{P}$ with Alice's encoding function $\rho : \{-1,1\}^n \to \mathrm{D}(\mathbb{C}^{2^m})$, where $\mathbb{C}^{2^m}$ is the state space of m-qubit and $\mathrm{D}(\mathbb{C}^{2^m})$ denotes the set of all quantum states in $\mathbb{C}^{2^m}$. Let $p_x := 1/2^n$, $p_\sigma := 1/n!$ and $p_b := 1/2$, then our hard distribution is

$$\Pr[x,b,\sigma,w] = p_x p_\sigma p_b \mathbf{1}[B_f(x,\sigma)\circ b^{\alpha n/t} = w]. \tag{3.76}$$

Conditioning on Bob's input (σ, w), from his perspective, Alice sends the message ρ_x with probability $\Pr[x|\sigma,w]$. His best strategy to determine b conditioning on his input (σ,w) is no more than the chance to distinguish between two subsets ρ_x selected according to b. In

other words, no more than the chance to distinguish between the following $\rho_1^{\sigma,w}$ and $\rho_{-1}^{\sigma,w}$, each appearing with probability $\Pr[b=1|\sigma,w]$ and $\Pr[b=-1|\sigma,w]$, respectively,

$$\rho_1^{\sigma,w} = \frac{\sum_{x\in\{-1,1\}^n} \Pr[x,1,\sigma,w]\rho_x}{\Pr[1,\sigma,w]} \text{ and } \rho_{-1}^{\sigma,w} = \frac{\sum_{x\in\{-1,1\}^n} \Pr[x,-1,\sigma,w]\rho_x}{\Pr[-1,\sigma,w]}. \tag{3.77}$$

Therefore, the bias ϵ_{bias} of any protocol $\mathscr{P}$ is the bias of distinguishing between $\rho_1^{\sigma,w}$ and $\rho_{-1}^{\sigma,w}$, which is given by the following result from [177]. For the sake of completeness we include the proof.

Lemma 3.12 ([177, Lemma 4.2]). *The bias ϵ_{bias} of the protocol $\mathscr{P}$ satisfies*

$$\epsilon_{bias} \leq \frac{1}{2} \sum_{S\subseteq[n]} \sum_{\sigma\in\mathbb{S}_n} \sum_{w\in\{-1,1\}^{\alpha n/t}} |u(\sigma,w,S)| \|\widehat{\rho}(S)\|_{\mathrm{tr}}, \tag{3.78}$$

where

$$u(\sigma,w,S) := \frac{1}{2} \sum_{x\in\{-1,1\}^n} p_x p_\sigma \chi_S(x) \left(\mathbf{1}\left[B_f(x,\sigma)=w\right] - \mathbf{1}\left[B_f(x,\sigma)=\overline{w}\right]\right). \tag{3.79}$$

Proof. The success probability of $\mathscr{P}$ given σ and w is, by Lemma 3.10,

$$\Pr[\mathscr{P} \text{ succeeds}|\sigma,w] \leq \frac{1}{2} + \frac{1}{2}\left\|\Pr[b=1|\sigma,w]\cdot\rho_1^{\sigma,w} - \Pr[b=-1|\sigma,w]\cdot\rho_{-1}^{\sigma,w}\right\|_{\mathrm{tr}}. \tag{3.80}$$

By taking the average over different inputs σ, w, we have

$$\Pr[\mathscr{P} \text{ succeeds}] \leq \frac{1}{2} + \frac{1}{2}\sum_{\sigma,w}\Pr[\sigma,w]\left\|\Pr[b=1|\sigma,w]\cdot\rho_1^{\sigma,w} - \Pr[b=-1|\sigma,w]\cdot\rho_{-1}^{\sigma,w}\right\|_{\mathrm{tr}}, \tag{3.81}$$

and thus,

$$\begin{aligned}
\epsilon_{bias} &\leq \frac{1}{2}\sum_{\sigma,w}\left\|\sum_x \Pr[x,1,\sigma,w]\cdot\rho_x - \Pr[x,-1,\sigma,w]\cdot\rho_x\right\|_{\mathrm{tr}} && (3.82)\\
&= \frac{1}{2}\sum_{\sigma,w}\left\|\sum_x \frac{1}{2}p_x p_\sigma \left(\mathbf{1}\left[B_f(x,\sigma)=w\right] - \mathbf{1}\left[B_f(x,\sigma)=\overline{w}\right]\right)\rho_x\right\|_{\mathrm{tr}} && (3.83)\\
&= \frac{1}{2}\sum_{\sigma,w}\left\|\sum_x \frac{1}{2}p_x p_\sigma \left(\mathbf{1}\left[B_f(x,\sigma)=w\right] - \mathbf{1}\left[B_f(x,\sigma)=\overline{w}\right]\right)\sum_S \widehat{\rho}(S)\chi_S(x)\right\|_{\mathrm{tr}} && (3.84)\\
&= \frac{1}{2}\sum_{\sigma,w}\left\|\sum_S u(\sigma,w,S)\widehat{\rho}(S)\right\|_{\mathrm{tr}} && (3.85)\\
&\leq \frac{1}{2}\sum_{S,\sigma,w}|u(\sigma,w,S)|\|\widehat{\rho}(S)\|_{\mathrm{tr}}, && (3.86)
\end{aligned}$$

where we used Eq. (3.76), the definition of $u(\sigma,w,S)$ and the Fourier expansion of ρ_x. ∎

We now need to analyse $|u(\sigma, w, S)|$ for different σ, w, S. The way it is done is by 'breaking' $\sigma(S)$ into blocks. Consider the set $[i|j] := \{i+1, i+2 \ldots, j\}$. Given $V \subseteq [n]$, we note that the set $V \cap [(j-1)t|jt]$ contains the elements of V that are in the interval $[(j-1)t+1, jt]$. From this we define $U_j \subseteq [t]$, for $j \in [n/t]$, as the set with elements from $V \cap [(j-1)t|jt]$ all shifted by $-(t-1)j$, so they fall in the interval $[1, t]$. It is clear that $V = \bigcup_{j=1}^{n/t} V \cap [(j-1)t|jt]$, which we shall write as $V = [n/t] \bullet \mathbb{U}_V$, where $\mathbb{U}_V = (U_1, \ldots, U_{n/t}) \in [t]^{n/t}$. The sequence of sets $\mathbb{U}_V$ is giving the decomposition of V into n/t blocks of length t. From it we can define $\mathbb{U}_V^*$ as the sequence of entries from $\mathbb{U}_V$ that are nonempty. With these in mind, the quantity $|u(\sigma, w, S)|$ is given by the following lemma, which is proved at the end of the section.

Lemma 3.13. *Given $S \subseteq [n]$ and $\sigma \in \mathbb{S}_n$, denote $\sigma(S) = [n/t] \bullet \mathbb{U}_{\sigma(S)}$, where $\mathbb{U}_{\sigma(S)} = (U_1, \ldots, U_{n/t}) \in [t]^{n/t}$. Define $\Delta = \{V \subseteq [\alpha n] \mid |\mathbb{U}_V^*|$ odd and $|U| \geq d,\ \forall U \in \mathbb{U}_V^*\}$, where $\mathrm{phdeg}(f) = d$. Then*

$$|u(\sigma, w, S)| = \begin{cases} \frac{p_\sigma}{2^{\alpha n/t}} \prod_{U \in \mathbb{U}^*_{\sigma(S)}} |\widehat{f}(U)| & \text{if } \sigma(S) \in \Delta, \\ 0 & \text{if } \sigma(S) \notin \Delta. \end{cases} \tag{3.87}$$

From Lemma 3.13 we see that $u(\sigma, w, S)$ is only nonzero when $\sigma(S) \in \Delta$, so the expression for the bias (Eq. (3.78)) becomes

$$\epsilon_{bias} \leq \frac{1}{2} \sum_{S \subseteq [n]} \sum_{\substack{\sigma \in \mathbb{S}_n \\ \sigma(S) \in \Delta}} p_\sigma \sum_{w \in \{-1,1\}^{\alpha n/t}} \frac{1}{2^{\alpha n/t}} \|\widehat{\rho}(S)\|_{\mathrm{tr}} \prod_{U \in \mathbb{U}^*_{\sigma(S)}} |\widehat{f}(U)| \tag{3.88}$$

$$= \frac{1}{2} \sum_{S \subseteq [n]} \|\widehat{\rho}(S)\|_{\mathrm{tr}} \sum_{\substack{\sigma \in \mathbb{S}_n \\ \sigma(S) \in \Delta}} p_\sigma \prod_{U \in \mathbb{U}^*_{\sigma(S)}} |\widehat{f}(U)|. \tag{3.89}$$

One of the requirements for $\sigma(S) \in \Delta$ is that the block decomposition of $\sigma(S)$ must have an odd number of nonempty blocks. Given S and σ, the number of nonempty blocks of $\sigma(S)$ (which we will denote by k below) is lower-bounded by $\lceil |S|/t \rceil \geq 1$ and upper-bounded by $\min(\lfloor |S|/d \rfloor, \alpha n/t) \leq \alpha n/t$ (since $\sigma(S) \subseteq [\alpha n]$). With these points in mind, we can write

$$\sum_{\substack{\sigma \in \mathbb{S}_n \\ \sigma(S) \in \Delta}} p_\sigma \prod_{U \in \mathbb{U}^*_{\sigma(S)}} |\widehat{f}(U)| \tag{3.90}$$

$$= \frac{1}{n!} \sum_{\sigma \in \mathbb{S}_n} [\sigma(S) \in \Delta] \prod_{U \in \mathbb{U}^*_{\sigma(S)}} |\widehat{f}(U)| \tag{3.91}$$

$$\leq \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \frac{\binom{\alpha n/t}{k}}{n!} \sum_{\substack{U_1, \ldots, U_k \subseteq [t] \\ d \leq |U_j| \leq t}} \mathbf{1}\left[\sum_{j=1}^k |U_j| = |S|\right] |\{\sigma \in \mathbb{S}_n \mid \sigma^{-1}([k] \bullet \mathbb{U}) = S\}| \prod_{j=1}^k |\widehat{f}(U_j)| \tag{3.92}$$

$$= \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \frac{\binom{\alpha n/t}{k}}{\binom{n}{|S|}} \sum_{\substack{U_1, \ldots, U_k \subseteq [t] \\ d \leq |U_j| \leq t}} \mathbf{1}\left[\sum_{j=1}^k |U_j| = |S|\right] \prod_{j=1}^k |\widehat{f}(U_j)|, \tag{3.93}$$

using that $|\{\sigma \in \mathbb{S}_n \mid \sigma^{-1}([k] \bullet \mathbb{U}) = S\}| = |S|!(n-|S|)!$. By plugging Eq. (3.93) into Eq. (3.89),

$$\epsilon_{bias} \leq \frac{1}{2} \sum_{S\subseteq[n]} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \frac{\binom{\alpha n/t}{k}}{\binom{n}{|S|}} \|\widehat{\rho}(S)\|_{\mathrm{tr}} \sum_{\substack{U_1,\dots,U_k\subseteq[t] \\ d\leq|U_j|\leq t}} \mathbf{1}\left[\sum_{j=1}^{k} |U_j| = |S|\right] \prod_{j=1}^{k} |\widehat{f}(U_j)| \tag{3.94}$$

$$\leq \frac{1}{2} \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \sum_{\substack{U_1,\dots,U_k\subseteq[t] \\ d\leq|U_j|\leq t}} \frac{\prod_{j=1}^{k} |\widehat{f}(U_j)|}{\binom{\alpha n/t}{k}} \sum_{\substack{S\subseteq[n] \\ |S|=\sum_{j=1}^{k}|U_j|}} \frac{\binom{\alpha n/t}{k}^2}{\binom{n}{|S|}} \|\widehat{\rho}(S)\|_{\mathrm{tr}}. \tag{3.95}$$

By Cauchy-Schwarz,

$$\sum_{\substack{S\subseteq[n] \\ |S|=\sum_{j=1}^{k}|U_j|}} \frac{\binom{\alpha n/t}{k}^2}{\binom{n}{|S|}} \|\widehat{\rho}(S)\|_{\mathrm{tr}} \leq \sqrt{\sum_{\substack{S\subseteq[n] \\ |S|=\sum_{j=1}^{k}|U_j|}} \frac{\binom{\alpha n/t}{k}^2}{\binom{n}{|S|}}} \sqrt{\sum_{\substack{S\subseteq[n] \\ |S|=\sum_{j=1}^{k}|U_j|}} \frac{\binom{\alpha n/t}{k}^2}{\binom{n}{|S|}} \|\widehat{\rho}(S)\|_{\mathrm{tr}}^2} \tag{3.96}$$

$$= \binom{\alpha n/t}{k} \sqrt{\sum_{\substack{S\subseteq[n] \\ |S|=\sum_{j=1}^{k}|U_j|}} \frac{\binom{\alpha n/t}{k}^2}{\binom{n}{|S|}} \|\widehat{\rho}(S)\|_{\mathrm{tr}}^2}. \tag{3.97}$$

By noticing that $kd \leq |S| \leq kt$ for a given k, and that $\binom{\alpha n/t}{k} \leq (\frac{\alpha|S|}{kt})^k \binom{kn/|S|}{k}$ and using the inequality $\binom{n}{m}^2 \binom{ln}{lm}^{-1} \leq (\frac{m}{n})^{(l-2)m}$ (see [177, Appendix A.5]), we get that

$$\frac{\binom{\alpha n/t}{k}^2}{\binom{n}{|S|}} \leq \left(\frac{\alpha|S|}{kt}\right)^{2k} \left(\frac{|S|}{n}\right)^{|S|-2k} \leq \alpha^{|S|} \left(\frac{|S|}{\alpha n}\right)^{|S|-2k} \leq \alpha^{kd} \left(\left(\frac{|S|}{\alpha n}\right)^{1-2k/|S|}\right)^{|S|} \tag{3.98}$$

$$\leq \alpha^{kd} \left(\left(\frac{kt}{\alpha n}\right)^{1-2/d}\right)^{|S|} = \alpha^{kd} \delta_k^{|S|}, \tag{3.99}$$

where we defined $\delta_k = (kt/\alpha n)^{1-2/d} \leq 1$. Thus we can apply the matrix-valued hypercontractive inequality, more specifically Theorem 1.31, giving

$$\sum_{\substack{S\subseteq[n] \\ |S|=\sum_{j=1}^{k}|U_j|}} \delta_k^{|S|} \|\widehat{\rho}(S)\|_{\mathrm{tr}}^2 \leq \sum_{S\subseteq[n]} \delta_k^{|S|} \|\widehat{\rho}(S)\|_{\mathrm{tr}}^2 \leq 2^{2\delta_k m}. \tag{3.100}$$

Putting Eqs. (3.97) and (3.100) into the bias expression (Eq. (3.95)), we finally get

$$2\epsilon_{bias} \leq \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} \alpha^{kd/2} 2^{\delta_k m} \sum_{\substack{U_1,\dots,U_k\subseteq[t] \\ d\leq|U_j|\leq t}} \prod_{j=1}^{k} |\widehat{f}(U_j)| = \sum_{\substack{k=1 \\ k \text{ odd}}}^{\alpha n/t} (\alpha^{d/2} \hat{\|} f \hat{\|}_1)^k 2^{\delta_k m}, \tag{3.101}$$

where we defined $\hat{\|}f\hat{\|}_1 := \sum_{U\subseteq[t]} |\widehat{f}(U)|$. By naming $\widetilde{\alpha} = \alpha^{d/2}\hat{\|}f\hat{\|}_1$ and using $\alpha \leq (2\hat{\|}f\hat{\|}_1)^{-2/d} \implies \widetilde{\alpha} \leq 1/2$, we can express the bias as

$$2\epsilon_{bias} \leq \sum_{\substack{k=1\\ k \text{ odd}}}^{\alpha n/t} \widetilde{\alpha}^k 2^{\delta_k m} = \widetilde{\alpha} 2^{\delta_1 m} + \sum_{\substack{k=3\\ k \text{ odd}}}^{\alpha n/t} \widetilde{\alpha}^{k/2}\widetilde{\alpha}^{k/2} 2^{\delta_k m} \tag{3.102}$$

$$\leq \widetilde{\alpha} 2^{\delta_1 m} + \left(\sum_{\substack{k=3\\ k \text{ odd}}}^{\alpha n/t} \widetilde{\alpha}^{k/2}\right) \max_{\substack{3\leq k\leq \alpha n/t\\ k \text{ odd}}} \widetilde{\alpha}^{k/2} 2^{\delta_k m} \leq \widetilde{\alpha} 2^{\delta_1 m} + \frac{\widetilde{\alpha}^{3/2}}{1-\widetilde{\alpha}} \max_{\substack{3\leq k\leq \alpha n/t\\ k \text{ odd}}} \widetilde{\alpha}^{k/2} 2^{\delta_k m}, \tag{3.103}$$

where $\delta_k = (kt/\alpha n)^{1-2/d}$. For sufficiently small distributional error (such that $2\epsilon_{bias} \geq 0.95$) and for $0 \leq \widetilde{\alpha} \leq 0.5$, we have either $\frac{\widetilde{\alpha}^{3/2}}{1-\widetilde{\alpha}} \widetilde{\alpha}^{k/2} 2^{\delta_k m} \geq \widetilde{\alpha}^{3/2}$ for some $k \geq 3$ or $\widetilde{\alpha} 2^{\delta_1 m} \geq 0.95 - \widetilde{\alpha}^{3/2} \geq \widetilde{\alpha}^{0.9}$. In the first case, we have $2^{\delta_k m} \geq (1-\widetilde{\alpha})\widetilde{\alpha}^{-k/2}$ and so

$$m = \Omega\left(\frac{k}{\delta_k} \log \frac{1}{\widetilde{\alpha}}\right) = \Omega\left(\left(\frac{\alpha}{t}\right)^{1-2/d} k^{2/d} \log\left(\frac{1}{\alpha^{d/2}\hat{\|}f\hat{\|}_1}\right) n^{1-2/d}\right) \tag{3.104}$$

for $k \geq 3$. In the second case, $2^{\delta_1 m} \geq \widetilde{\alpha}^{-0.1}$ and thus

$$m = \Omega\left(\frac{1}{\delta_1} \log \frac{1}{\widetilde{\alpha}}\right) = \Omega\left(\left(\frac{\alpha}{t}\right)^{1-2/d} \log\left(\frac{1}{\alpha^{d/2}\hat{\|}f\hat{\|}_1}\right) n^{1-2/d}\right). \tag{3.105}$$

This concludes the proof. ■

Proof of Lemma 3.13. We start with the following:

$$\sum_{x\in\{-1,1\}^n} \chi_S(x)\mathbf{1}\big[B_f(x,\sigma) = w\big] = \sum_{x\in\{-1,1\}^n} \chi_S(x) \prod_{j=1}^{\alpha n/t} \mathbf{1}\left[f(\sigma(x)^{(j)}) = w_j\right] \tag{3.106}$$

$$= \sum_{x\in\{-1,1\}^n} \chi_{\sigma(S)}(\sigma(x)) \prod_{j=1}^{\alpha n/t} \mathbf{1}\left[f(\sigma(x)^{(j)}) = w_j\right] \tag{3.107}$$

$$= \sum_{x\in\{-1,1\}^n} \chi_{\sigma(S)}(x) \prod_{j=1}^{\alpha n/t} \mathbf{1}\left[f(x^{(j)}) = w_j\right]. \tag{3.108}$$

By using the block decomposition $\sigma(S) = [n/t] \bullet \mathbb{U}_{\sigma(S)}$ with $\mathbb{U}_{\sigma(S)} = (U_1, \ldots, U_{n/t}) \in [t]^{n/t}$,

$$\sum_{x\in\{-1,1\}^n} \chi_S(x)\mathbf{1}\big[B_f(x,\sigma) = w\big] = \sum_{x\in\{-1,1\}^n} \chi_{[n/t]\bullet\mathbb{U}_{\sigma(S)}}(x) \prod_{j=1}^{\alpha n/t} \mathbf{1}\left[f(x^{(j)}) = w_j\right] \tag{3.109}$$

$$= \sum_{x\in\{-1,1\}^n} \left(\prod_{i=1}^{n/t} \chi_{U_i}(x^{(i)})\right) \left(\prod_{j=1}^{\alpha n/t} \mathbf{1}\left[f(x^{(j)}) = w_j\right]\right) \tag{3.110}$$

$$= 2^{n(1-\alpha)} \sum_{x\in\{-1,1\}^{\alpha n}} \prod_{j=1}^{\alpha n/t} \chi_{U_j}(x^{(j)})\mathbf{1}\left[f(x^{(j)}) = w_j\right], \tag{3.111}$$

if $U_j = \emptyset$ for $\alpha n/t < j \leq n/t$, i.e., $\sigma(S) \subseteq [\alpha n]$; otherwise, the sum evaluates to 0.

$$\sum_{x\in\{-1,1\}^n} \chi_S(x)\mathbf{1}\big[B_f(x,\sigma) = w\big] = 2^{n(1-\alpha)} \prod_{j=1}^{\alpha n/t} \left(\sum_{x\in\{-1,1\}^t} \chi_{U_j}(x)\mathbf{1}\left[f(x) = w_j\right] \right) \tag{3.112}$$

$$= 2^{n(1-\alpha)} \prod_{j=1}^{\alpha n/t} \left(\sum_{x\in\{-1,1\}^t} \chi_{U_j}(x)\frac{1+f(x)w_j}{2} \right) \tag{3.113}$$

$$= \frac{2^n}{2^{\alpha n/t}} \prod_{j=1}^{\alpha n/t} \left(\mathbf{1}[U_j = \emptyset] + \widehat{f}(U_j)w_j \right). \tag{3.114}$$

Since $\mathrm{phdeg}(f) > 0$, $\widehat{f}(\emptyset) = 0$. Hence

$$\sum_{x\in\{-1,1\}^n} \chi_S(x)\mathbf{1}\big[B_f(x,\sigma) = w\big] = \frac{2^n}{2^{\alpha n/t}} \prod_{\substack{j\in[\alpha n/t] \\ U_j\neq\emptyset}} \widehat{f}(U_j)w_j. \tag{3.115}$$

By the same token,

$$\sum_{x\in\{-1,1\}^n} \chi_S(x)\mathbf{1}\big[B_f(x,\sigma) = \overline{w}\big] = \frac{2^n}{2^{\alpha n/t}} \prod_{\substack{j\in[\alpha n/t] \\ U_j\neq\emptyset}} \widehat{f}(U_j)\overline{w_j} = (-1)^{|\mathbb{U}^*_{\sigma(S)}|}\frac{2^n}{2^{\alpha n/t}} \prod_{\substack{j\in[\alpha n/t] \\ U_j\neq\emptyset}} \widehat{f}(U_j)w_j. \tag{3.116}$$

Therefore

$$u(\sigma,w,S) = \frac{p_\sigma}{2^{\alpha n/t}}\frac{1-(-1)^{|\mathbb{U}^*_{\sigma(S)}|}}{2} \prod_{\substack{j\in[\alpha n/t] \\ U_j\neq\emptyset}} \widehat{f}(U_j)w_j. \tag{3.117}$$

We see that $u(\sigma,w,S)$ can only be nonzero if $\sigma(S) \subseteq [\alpha n]$, if $|\mathbb{U}^*_{\sigma(S)}|$ is odd and if $d \leq |U| \leq t$ for all $U \in \mathbb{U}^*_{\sigma(S)}$, i.e., if $\sigma(S) \in \Delta$. In summary,

$$|u(\sigma,w,S)| = \begin{cases} \frac{p_\sigma}{2^{\alpha n/t}} \prod_{U\in\mathbb{U}^*_{\sigma(S)}} |\widehat{f}(U)| & \text{if } \sigma(S) \in \Delta, \\ 0 & \text{if } \sigma(S) \notin \Delta. \end{cases} \tag{3.118}$$

■

3.6 Limitations of proof technique

In the last section we saw that Theorem 3.8 guarantees the hardness of the f-BHP$_n^{\alpha,t}$ problem if f has pure high degree ≥ 2, but the hardness result is not guaranteed if only sign degree ≥ 2. To arrive at this result, we used the uniform distribution as a 'hard' distribution for Yao's principle. In this section we shall prove that under the uniform distribution we cannot obtain a better result. More specifically, we shall prove that under the uniform distribution there is an efficient bounded-error classical protocol for solving the f-BHP$_n^{\alpha,t}$ problem if $\mathrm{phdeg}(f) \leq 1$.

Theorem 3.14. *Under the uniform distribution for Alice and Bob's inputs, if* $\mathrm{phdeg}(f) \leq 1$*, then* $R^1(f\text{-BHP}_n^{\alpha,t}) = O(\log n)$ *for a success probability strictly greater than* $1/2$ *independent of* n*.*

Proof. Assume that f is non-constant, otherwise the result holds trivially. Let $F := \{i \in [t] \mid \widehat{f}(\{i\}) \neq 0\}$. Given that $\mathrm{phdeg}(f) \leq 1$, this set is non-empty. Consider the following protocol: Alice picks a subset $I \subseteq [n]$ of indices uniformly at random using shared randomness, where $|I|$ will be determined later, and sends the indices and corresponding bitvalues to Bob. Let $\{x_i\}_{i\in I}$ be the bitvalues sent, and let $j(i) = \lceil \sigma(i)/t \rceil$ and $k(i) \equiv \sigma(i) \pmod t$ for all $i \in I$, where $\sigma \in \mathbb{S}_n$ is Bob's permutation. The probability that none of the indices sent by Alice are matched to a non-zero Fourier coefficient according to Bob's permutation, within one of the $\alpha n/t$ blocks he has, is

$$\Pr_\sigma[k(i) \notin F,\ \forall i \in I] \leq \left(1 - \alpha\frac{|F|}{t}\right)^{|I|} \leq e^{-\alpha|I||F|/t}, \tag{3.119}$$

which we can make almost arbitrarily small by choosing $|I|$ to be sufficiently large. (Note that the first inequality above would be an equality if we chose the elements of I with replacement, and choosing them without replacement cannot make $\Pr[k(i) \notin F,\ \forall i \in I]$ higher). Hence, with high probability, $\exists i \in I \cap [\alpha n/t]$ such that $k(i) \in F$. Bob computes $\mathrm{sgn}[\widehat{f}(\{k(i)\})] \cdot \sigma(x)_{k(i)}^{(j(i))} \cdot w_{j(i)}$: if it is $+1$, he outputs that $B_f(x,\sigma) = w$, and if it is -1, he outputs that $B_f(x,\sigma) = \overline{w}$. Otherwise, if $k(i) \notin F$ for all $i \in I \cap [\alpha n/t]$, then Bob outputs a random bit.

To see why the protocol works, we calculate the probability that $\mathrm{sgn}[\widehat{f}(\{k(i)\})] \cdot \sigma(x)_{k(i)}^{(j(i))}$ is equal to $f(\sigma(x)^{(j(i))})$.

$$\Pr_x\left[\mathrm{sgn}[\widehat{f}(\{k(i)\})] \cdot \sigma(x)_{k(i)}^{(j(i))} = f(\sigma(x)^{(j(i))})\right] \tag{3.120}$$

$$= \frac{1}{2} + \frac{1}{2^{t+1}} \sum_{x\in\{-1,1\}^t} \mathrm{sgn}[\widehat{f}(\{k(i)\})] \cdot \sigma(x)_{k(i)}^{(j(i))} \cdot f(\sigma(x)^{(j(i))}) \tag{3.121}$$

$$= \frac{1}{2} + \frac{1}{2}\,\mathrm{sgn}[\widehat{f}(\{k(i)\})] \cdot \widehat{f}(\{k(i)\}) \tag{3.122}$$

$$= \frac{1}{2} + \frac{1}{2}|\widehat{f}(\{k(i)\})|, \tag{3.123}$$

which is greater than $1/2$ and where we used in the first line that the distribution on Alice's inputs is uniform. The overall success probability of the protocol ($\exists i \in I \cap [\alpha n/t]$ such that $k(i) \in F$ and Bob's output equals f) is at least $\frac{1}{2} + \frac{1}{2}|\widehat{f}(\{k(i)\})|(1 - e^{-\alpha|I||F|/t})$. By taking $|I| = O(1)$, this is strictly greater than $1/2$ by $\Omega(\frac{\alpha}{t}|\widehat{f}(\{k(i)\})|)$, which does not depend on n, since $|\widehat{f}(\{k(i)\})| \geq 2^{1-t}$ (as it is nonzero and is an average of 2^t ± 1's). ■

3.7 Conjectures

It is known that $\mathrm{phdeg}(f) \leq \mathrm{sdeg}(f)$ [176, 43], but our lower bounds are probably not tight for *all* functions with sign degree ≥ 2. We proved that this is an inherent limitation of the chosen

distribution for Alice and Bob's inputs during the proof, since under the uniform distribution it is possible to solve the problem with $O(\log n)$ bits of communication if $\mathrm{phdeg}(f) \leq 1$. We then make the following conjectures.

Conjecture 3.15. $R^1_\epsilon(f\text{-BHP}^{\alpha,t}_n) = \Omega(n^{1-1/d})$ *if* $\mathrm{sdeg}(f) = d \geq 2$.

Conjecture 3.16. $Q^1_\epsilon(f\text{-BHP}^{\alpha,t}_n) = \Omega(n^{1-2/d})$ *if* $\mathrm{sdeg}(f) = d \geq 3$.

A proof of these results would require a non-uniform distribution on Alice and Bob's inputs, as proven in Theorem 3.14.

Chapter 4

Quantum Random Access Codes for Boolean Functions

A quantum random access code (QRAC) is an encoding of a number of bits into a smaller number of qubits such that any one of the initial bits can be recovered with some probability of success. A QRAC is normally denoted by $n \overset{p}{\mapsto} m$, meaning that n bits are encoded into m qubits such that any initial bit can be recovered with probability at least $p > 1/2$ (greater than $1/2$ since $p = 1/2$ can be achieved by pure guessing), and a classical version, called simply *random access code* (RAC), is similarly defined, with the encoding message being m bits. The idea of QRACs first appeared in a paper by Stephen Wiesner [198] in 1983 under the name of *conjugate coding*, and was later rediscovered by Ambainis *et al.* in 1999 [12].

Quantum random access codes found application in many different contexts, e.g. quantum finite automata [12, 150], network coding [96, 97], quantum communication complexity [42, 82, 119], locally decodable codes [24, 114, 115, 197], non-local games [149, 191], cryptography [164], quantum state learning [1], device-independent dimension witnessing [4, 7, 196], self-testing measurements [72, 73], randomness expansion [132], studies of no-signaling resources [91], and characterization of quantum correlations from information theory [165]. The $2 \mapsto 1$ and $3 \mapsto 1$ QRACs were first experimentally demonstrated in [183]. See [78, 94, 149, 190, 194] for subsequent demonstrations.

In this chapter we further generalise the idea of (quantum) random access codes to recovering not just an initial bit, but the value of a fixed Boolean function on any subset of the initial bits with fixed size. We call them f-random access codes. The case of the function Parity was already considered in [24], and here we generalise to arbitrary Boolean functions f.

4.1 Related Work

An $n \overset{p}{\mapsto} m$ (Q)RAC is an encoding of n bits into m (qu)bits such that any initial bit can be recovered with probability at least p. This probability is the *worst case success probability* over all possible pairs (x, i) of input string $x \in \{-1, 1\}^n$ and recoverable bit $i \in \{1, \ldots, n\}$. Many different resources can be used during the encoding and decoding, e.g. private randomness (PR), shared randomness (SR), shared entanglement, and even super-quantum correlations like Popescu-Rohrlich boxes [169].

Regarding the classical RAC, Ambainis *et al.* [12] proved that there is no $2 \overset{p}{\mapsto} 1$ RAC (and $2^m \overset{p}{\mapsto} m$ RAC by extension) with PR and worst case success probability $p > 1/2$. On the other hand, Ambainis *et al.* [11] showed that RACs with SR can achieve success probability $p > 1/2$.

Theorem 4.1 ([11, Equation (25)])**.** *The optimal $n \overset{p}{\mapsto} 1$ RAC with SR has success probability*

$$p = \frac{1}{2} + \frac{1}{2^n}\binom{n-1}{\lfloor \frac{n-1}{2} \rfloor} = \frac{1}{2} + \frac{1}{\sqrt{2\pi n}} - O\left(\frac{1}{n^{3/2}}\right). \tag{4.1}$$

For a general number of encoded bits, Ambainis *et al.* [12] developed a RAC with PR using a specific code from [50] which matches their classical lower bound $m \geq (1 - H(p))n$ up to an additive logarithmic term, where $H(p) = -p \log_2 p - (1-p)\log_2(1-p)$ is the binary entropy function.

Theorem 4.2 ([12, Theorem 2.2])**.** *There is an $n \overset{p}{\mapsto} m$ RAC with PR and $m \leq (1 - H(p))n + 7 \log n$ for any $p > \frac{1}{2}$.*

As for QRACs, Ambainis *et al.* [12] showed the existence of a $2 \overset{p}{\mapsto} 1$ QRAC with PR[1] and $p = \frac{1}{2} + \frac{1}{2\sqrt{2}} \approx 0.85$, and the existence of a $3 \overset{p}{\mapsto} 1$ QRAC with PR and $p = \frac{1}{2} + \frac{1}{2\sqrt{3}} \approx 0.79$ (the second attributed to Chuang). Later Hayashi *et al.* [96] showed the impossibility of a $4 \overset{p}{\mapsto} 1$ QRAC (and $4^m \overset{p}{\mapsto} m$ QRAC by extension) with PR and success probability $p > 1/2$. Similarly to the classical case, Ambainis *et al.* [11] also showed that QRACs can benefit from SR.

Theorem 4.3 ([11, Theorem 6])**.** *There is an $n \overset{p}{\mapsto} 1$ QRAC with SR and* [2]

$$p = \frac{1}{2} + \sqrt{\frac{2}{3\pi n}} + O\left(\frac{1}{n^{3/2}}\right). \tag{4.2}$$

The specific case of $m = 2$ encoding qubits was explored in [96, 107, 133]. For the general case of $m > 1$, Iwama *et al.* [108] constructed an $(4^m - 1) \overset{p}{\mapsto} m$ QRAC with PR and $p = \frac{1}{2} + \frac{1}{2(2^m-1)\sqrt{2^m+1}}$ (such construction also works for all $n < 4^m$). On the other hand, Ambainis *et al.* [12] proved that if an $n \overset{p}{\mapsto} m$ QRAC with PR and $p > 1/2$ exists, then

[1] Usually private randomness is already assumed in QRACs under the encoding onto density matrices.

[2] Ambainis *et al.* [11] do not give the high order terms, but these can be calculated by following their procedure together with [105, Equation 2.198].

$m = \Omega(n/\log n)$, which was later improved to $m \geq (1 - H(p))n$ by Nayak [150], thus matching the same classical lower bound from [12].

The idea of decoding a function of the initial bits instead of a single bit was already considered by Ben-Aroya, Regev and de Wolf [24] (who also considered recovering multiple bits rather than just one). More specifically, they defined an $n \overset{p}{\mapsto} m$ k-XOR-QRAC, where n bits are encoded into m qubits such that the Parity of any k initial bits can be recovered with success probability at least p.[3] Using their hypercontractive inequality for matrix-valued functions, they proved the following upper bound on the success probability.

Theorem 4.4 ([24, Theorem 7])**.** *For any $\eta > 2\ln 2$ there is a constant C_η such that, if n/k is large enough, then for any $n \overset{p}{\mapsto} m$ k-XOR-QRAC with SR,*

$$p \leq \frac{1}{2} + C_\eta \left(\frac{\eta m}{n}\right)^{k/2}. \tag{4.3}$$

They conjectured that the factor $\eta > 2\ln 2$ can be dropped from the above bound, and thus extended to $m/n > 1/(2\ln 2) \approx 0.72$, although it might require a strengthening of their hypercontractive inequality.

The use of shared entanglement in random access codes was first considered by Klauck [119, 121]. Here the encoding and decoding parties are allowed to use an arbitrary amount of shared entangled states.[4] The figure of merit in this generalisation is the relation between n, m and p, while the amount of shared entanglement is not taken into account. Klauck [119, 121] considered an $n \overset{p}{\mapsto} m$ QRAC with shared entanglement and, by its equivalence to the quantum one-way communication complexity for the index function, proved the lower bound $m \geq (1 - H(p))n/2$, similar to Nayak's bound. Later Pawłowski and Żukowski [166] coined the term entanglement-assisted random access code (EARAC), which is a RAC with shared entanglement, and studied the case when $m = 1$, giving protocols with better decoding probabilities compared to the usual $n \overset{p}{\mapsto} 1$ QRAC with SR. Recently Tănăsescu *et al.* [189] expanded the idea of $n \overset{p}{\mapsto} 1$ EARACs to recovering an initial bit under a specific request distribution.

Theorem 4.5 ([166] and [189, Corollary 2 and Theorem 5])**.** *The optimal $n \overset{p}{\mapsto} 1$ EARAC with SR has success probability*

$$p = \frac{1}{2} + \frac{1}{2\sqrt{n}}. \tag{4.4}$$

The idea of (Q)RAC was generalised in other ways, e.g. parity-oblivious [9, 47, 183] and multiparty [172] versions, encoding on d-valued qubits (qudits) [10, 46, 73, 133, 190], a wider range of information retrieval tasks [69] and a connection to Popescu-Rohrlich boxes. It was shown [200] that a Popescu-Rohrlich box can simulate a RAC by means of just one bit of

[3] In their definition the success probability is the average over random k subsets and random inputs, which, in our context, is equivalent to using SR.

[4] We note that shared entanglement can be used to obtain both private and shared randomness.

communication, while in [91] the converse was proved. An object called *racbox* [91] was defined, which is a box that implements a RAC when supported with one bit of communication, and it was shown that a *non-signaling* racbox is equivalent to a Popescu-Rohrlich box. A quantum version of a racbox was later proposed in [92]. Finally, we mention that RACs were also studied within "theories" that violate the uncertainty relation for anti-commuting observables and present stronger-than-quantum correlations [187].

4.1.1 Our Results

This chapter focuses on generalizing the classical, quantum and entanglement-assisted random access codes. Instead of recovering a single bit from the initial string $x \in \{-1,1\}^n$, we are interested in evaluating a Boolean function $f : \{-1,1\}^k \to \{-1,1\}$ on any k sequence of bits from x. We generically call them f-random access codes. Let $\mathcal{S}_n^k = \{(S_i)_{i=1}^k \in \{1,\dots,n\}^k \mid S_i \neq S_j \ \forall i,j\}$ be the set of sequences of different elements from $\{1,\dots,n\}$ with length k and let $x_S \in \{-1,1\}^k$ denote the substring of $x \in \{-1,1\}^n$ specified by $S \in \mathcal{S}_n^k$. Alice gets $x \in \{-1,1\}^n$ and she needs to encode her data and send it to Bob, so that he can decode $f(x_S)$ for any $S \in \mathcal{S}_n^k$ with probability $p > 1/2$. Such problem was already considered by Sherstov in a *two-way communication complexity* setting [175] and later used in his pattern matrix method [176] in order to prove other communication complexity lower bounds. Even though our results are expressed in a random access code language, they can also be seen as in a *one-way communication complexity* setting. If two-way communication is allowed, Bob can send the identity of his sequence to Alice with $O(k \log n)$ bits of communication, whereas (as we will see) significantly more communication may be required in the one-way scenario.

In the following, Π will refer to a sample space with some probability distribution. As before, PR and SR stand for private and shared randomness, respectively. Moreover, since we require the success probability to always be greater than $1/2$, given that one can always guess the correct result with probability $1/2$, from now on it will be convenient to use the *bias* ε of the prediction, defined as $\varepsilon = 2p - 1$, instead of its success probability p (the bias from this chapter differs from the bias from Chapter 3 by a factor of 2).

We define $n \overset{\varepsilon}{\mapsto} m$ f-RAC, the f-classical random access code on m bits with bias ε.

Definition 4.6. An $n \overset{\varepsilon}{\mapsto} m$ f-RAC with PR is an encoding map $E : \{-1,1\}^n \times \Pi_A \to \{-1,1\}^m$ satisfying the following: for every $S \in \mathcal{S}_n^k$ there is a decoding map $D_S : \{-1,1\}^m \times \Pi_B \to \{-1,1\}$ such that $\Pr_{r_A,r_B}[D_S(E(x,r_A),r_B) = f(x_S)] \geq \frac{1}{2} + \frac{1}{2}\varepsilon$ for all $x \in \{-1,1\}^n$.

Definition 4.7. An $n \overset{\varepsilon}{\mapsto} m$ f-RAC with SR is an encoding map $E : \{-1,1\}^n \times \Pi \to \{-1,1\}^m$ satisfying the following: for every $S \in \mathcal{S}_n^k$ there is a decoding map $D_S : \{-1,1\}^m \times \Pi \to \{-1,1\}$ such that $\Pr_r[D_S(E(x,r),r) = f(x_S)] \geq \frac{1}{2} + \frac{1}{2}\varepsilon$ for all $x \in \{-1,1\}^n$.

We define $n \overset{\varepsilon}{\mapsto} m$ f-QRAC, the f-quantum random access code on m qubits with bias ε.

Definition 4.8. An $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with PR is an encoding map $E : \{-1,1\}^n \to \mathbb{C}^{2^m \times 2^m}$ that assigns an m-qubit density matrix to every $x \in \{-1,1\}^n$ and satisfies the following: for every $S \in \mathcal{S}_n^k$ there is a POVM $M^S = \{M_{-1}^S, M_1^S\}$ such that $\mathrm{Tr}(M_{f(x_S)}^S \cdot E(x)) \geq \frac{1}{2} + \frac{1}{2}\varepsilon$ for all $x \in \{-1,1\}^n$.

Definition 4.9. An $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with SR is an encoding map $E : \{-1,1\}^n \times \Pi \to \mathbb{C}^{2^m}$ that assigns an m-qubit pure state to every $x \in \{-1,1\}^n$ and satisfies the following: for every $S \in \mathcal{S}_n^k$ there is a set of POVMs $\{M_r^S\}_{r\in\Pi}$, with $M_r^S = \{M_{-1,r}^S, M_{1,r}^S\}$, such that $\mathbb{E}_r[E(x,r)^\dagger M_{f(x_S),r}^S E(x,r)] \geq \frac{1}{2} + \frac{1}{2}\varepsilon$ for all $x \in \{-1,1\}^n$.

Similarly, we define $n \overset{\varepsilon}{\mapsto} m$ f-EARAC, the f-entanglement-assisted random access code on m bits with bias ε.

Definition 4.10. An $n \overset{\varepsilon}{\mapsto} m$ f-EARAC is an $n \overset{\varepsilon}{\mapsto} m$ f-RAC with SR where the encoding and decoding parties share an unlimited amount of entangled quantum states.

Due to shared entanglement being a source of SR, we already include SR in f-EARACs. We note that [166] focused on EARACs without SR.

Finally, we define $n \overset{\varepsilon}{\mapsto} m$ f-PRRAC, the f-Popescu-Rohrlich random access code on m bits with bias ε. A Popescu-Rohrlich box [169] is a bipartite system shared by two parties with two inputs $x, y \in \{0,1\}$ and two outputs $a, b \in \{0,1\}$ and is defined by the joint probability distribution

$$\Pr[ab|xy] = \begin{cases} \frac{1}{2} & \text{for } a \oplus b = xy, \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Definition 4.11. An $n \overset{\varepsilon}{\mapsto} m$ f-PRRAC is an $n \overset{\varepsilon}{\mapsto} m$ f-RAC with SR where the encoding and decoding parties share an unlimited amount of Popescu-Rohrlich boxes.

In Section 3.3 we devise encoding-decoding strategies for all the f-random access codes just defined, thus deriving *lower bounds* on their biases given the encoding/decoding parameters n, m and k. These f-random access codes are built based on previous ideas from [11, 12, 166]. The Boolean function that needs to be evaluated directly influences the final bias and such influence in our results is captured by the single quantity called *noise stability* (see Section 1.2).

Our positive results can be summarized by the following theorem.

Theorem 4.12. *Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function and $\mathrm{Stab}_q[f]$ its noise stability with parameter q.*

(a) Let $\ell \in \mathbb{N}$. If $m = \Omega(\ell \log n)$ and $k = o(\sqrt{\ell})$, there is an $n \overset{\varepsilon}{\mapsto} m$ f-RAC with PR and bias $\varepsilon \geq (1 - o_n(1))\,\mathrm{Stab}_q[f]$ with $q \geq \sqrt{\frac{m}{n} - \frac{5\log_2(n/\ell)}{n/\ell}}$.

(b) *If* $k = o(\sqrt{m})$, *there is an* $n \overset{\varepsilon}{\mapsto} m$ f*-RAC with SR and* $\varepsilon \geq (1 - o_n(1)) \operatorname{Stab}_q[f]$ *with* $q \geq \sqrt{\frac{m}{2n}}$.

(c) *If* $k = o(\sqrt{m})$, *there is an* $n \overset{\varepsilon}{\mapsto} m$ f*-QRAC with SR and* $\varepsilon \geq (1 - o_n(1)) \operatorname{Stab}_q[f]$ *with* $q \geq \sqrt{\frac{8m}{3\pi n}}$.

(d) *If* $k = o(\sqrt{m})$, *there is an* $n \overset{\varepsilon}{\mapsto} m$ f*-EARAC with* $\varepsilon \geq (1 - o_n(1)) \operatorname{Stab}_q[f]$ *and* $q = \sqrt{\frac{m}{n}}$.

(e) *For any* $n \in \mathbb{N}$, *there is an* $n \overset{1}{\mapsto} 1$ f*-PRRAC.*

Results (a), (b), (c) and (d) use an encoding scheme reminiscent of the concatenation idea from [165, 166, 189] (and suggested to us by Ronald de Wolf). The underlying idea is to randomly break the initial string $x \in \{-1, 1\}^n$ into different 'blocks' and encode them via a standard RAC/QRAC/EARAC. Result (a) breaks x into ℓ blocks and employs the $n/\ell \mapsto m/\ell$ RAC from Theorem 4.2 on every block, each with n/ℓ elements, while in results (b)/(c)/(d) we employ the $n/m \mapsto 1$ RAC/QRAC/EARAC from Theorems 4.1/4.3/4.5 in order to encode m blocks, each with n/m elements, into a single (qu)bit each, resulting in m encoded (qu)bits. With high probability all the bits from the needed string $x_S \in \{-1, 1\}^k$ will be encoded into different blocks and therefore can be decoded and f evaluated. The decoded string $y \in \{-1, 1\}^k$ can be viewed as a 'noisy' x_S, to which the noise stability framework can be applied. The bias of the base RAC/QRAC/EARAC thus becomes the parameter q in the noise stability of the corresponding f-random access code. As a quick remark, since we opted to lower-bound the parameters q in Theorem 4.12, in result (b) q does not exactly equal the bias from Theorem 4.1. One could write, though, $q \approx \sqrt{\frac{2m}{\pi n}}$.

Result (a) is our strongest bound, since it also applies to all other f-random access codes. Moreover, there is some freedom in setting the number of blocks ℓ, since the number of encoded bits in Theorem 4.2 is not fixed to a single number (as opposed to Theorems 4.1, 4.3 and 4.5). The result is a trade-off between the number of bits k of the Boolean function and the number of encoded bits m. However, the number of encoded bits in result (a) is limited to $m = \Omega(\log n)$, a characteristic inherited from the RAC in Theorem 4.2. It is possible to go below this limit by using SR, as demonstrated by results (b), (c) and (d).

The above results show that quantum resources offer a modest advantage over the classical f-random access code. On the other hand, result (e) demonstrates that stronger-than-quantum resources like Popescu-Rohrlich boxes can lead to extremely powerful f-random access codes. This is a consequence of violating Information Causality [165], since one bit transfer allows the access to *any* bit in a database via Popescu-Rohrlich boxes. From $x \in \{-1, 1\}^n$ a long bit-string $x_f \in \{-1, 1\}^t$, where $t = |\mathcal{S}_n^k|$, can be constructed with the values $f(x_S)$ for all $S \in \mathcal{S}_n^k$. All bits from x_f are readable with the aid of Popescu-Rohrlich boxes, with non-signaling constraining the readout to just one bit. The protocol for f-PRRACs is taken from [165] and uses a pyramid of Popescu-Rohrlich boxes and nests a van Dam's protocol [55].

In Section 4.3 we prove an *upper bound* on the bias of any f-QRAC with SR (and f-RAC) using the same method of the hypercontractive inequality for matrix-valued functions from [24].

Theorem 4.13. *Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. For any $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with SR and $k = o(n)$ the following holds: for any $\eta > 2\ln 2$ there is a constant C_η such that*

$$\varepsilon \leq C_\eta \sum_{\ell=0}^{k} L_{1,\ell}(f) \left(\frac{\eta m}{n}\right)^{\ell/2}, \tag{4.6}$$

where $L_{1,\ell}(f) = \sum_{\substack{T\subseteq[k]\\|T|=\ell}} \left|\widehat{f}(T)\right|$ is the 1-norm of the ℓ-th level of the Fourier transform of f.

One can see that the above result is a generalisation of Theorem 4.4. Indeed, for Parity on k bits, $L_{1,\ell}(\mathrm{XOR}_k) = 1$ iff $\ell = k$, and so Eq. (4.3) is recovered. The following corollary from Theorem 4.13 helps to compare the bias upper bound to the bias lower bounds from Theorem 4.12.

Corollary 4.14. *Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. For any $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with SR and $k = o(n)$ the following holds: for any $\eta > 2\ln 2$ there is a constant C_η such that*

$$\varepsilon \leq C_\eta 2^{\deg(f)-1} \mathrm{Stab}_q[f], \tag{4.7}$$

where $q = \sqrt{\frac{\eta m}{n}}$ and $\deg(f) = \max\{|S| : \widehat{f}(S) \neq 0\}$ is the degree of f.

Taking $\deg(f)$ to be upper-bounded by a constant (for example, if $k = O(1)$), our bias upper bound matches our bias lower bounds for f-RAC/QRAC with SR up to a global multiplicative constant and a multiplicative constant $\sqrt{\eta}$ in the parameter q. We conjecture that the parameter q can be improved to $\sqrt{\frac{m}{n}}$, which might require a stronger version of the hypercontractive inequality for matrix-valued functions or some other approach. Other corollaries from Theorem 4.13 are derived in Section 4.3 and compared to our bias lower bounds.

Upper bound (4.6) does not apply to f-EARACs. Previously, it was known that for the special case of standard EARACs ($m = 1$), the bias ε is upper-bounded by $1/\sqrt{n}$ (Theorem 4.5). This upper bound can generalised to EARACs with $m > 1$ assuming an independence condition (Section 4.2.4). The bound we obtain is $\varepsilon \leq \sqrt{m/n}$. We view this as evidence that the bias lower bound for the general case of f-EARACs given in Theorem 4.12 should actually be tight.

In Section 4.2 we present protocols for all our f-random access codes, and in Section 4.3 we derive an upper bound on the bias of f-QRACs with SR and a partial upper bound on the bias of EARACs.

4.2 Bias Lower Bounds

In the following, we write $[n] = \{1, \ldots, n\}$ and $\mathbb{S}_n$ is the set of all permutations of $[n]$. As before, let $\mathcal{S}_n^k = \{(S_i)_{i=1}^k \in [n]^k \mid S_i \neq S_j \ \forall i, j\}$ be the set of sequences of different elements from $[n]$ with length k and let $x_S \in \{-1,1\}^k$ denote the substring of $x \in \{-1,1\}^n$ specified by $S \in \mathcal{S}_n^k$.

4.2.1 f-RAC with PR

We start by studying the f-RAC with PR. The following result is based on Ambainis *et al.* [12] and uses a procedure reminiscent of the concatenation idea from [165, 166, 189]: the initial string is broken in blocks, which in turn are encoded using the code from [50]. First we state an useful bound on the binary entropy function $H(p) = -p\log_2 p - (1-p)\log_2(1-p)$.

Theorem 4.15 ([44, Theorem 2.2])**.** $\forall p \in [0,1]$, $1 - 4\left(p - \frac{1}{2}\right)^2 \leq H(p) \leq 1 - \frac{2}{\ln 2}\left(p - \frac{1}{2}\right)^2$.

We also state a slightly modified version of Newman's Theorem [152] (see Theorem 1.9) which is going to be useful to us.

Theorem 4.16 ([152])**.** *Let $\mathcal{E}(x,r)$ be an event of $x \in \{-1,1\}^n \times \{-1,1\}^n$ and $r \in R$ such that*

$$\Pr_{r\sim R}[\mathcal{E}(x,r)] \geq p \tag{4.8}$$

for all $x \in \{-1,1\}^n \times \{-1,1\}^n$, with $p \in (0,1]$. Let $\delta \in (0,p]$. Then there is $R_0 \subseteq R$ with size at most n/δ^2 such that

$$\Pr_{r\sim R_0}[\mathcal{E}(x,r)] \geq p - \delta \tag{4.9}$$

holds for all $x \in \{-1,1\}^n \times \{-1,1\}^n$.

Theorem 4.17. *Let $\ell \in \mathbb{N}$, $\ell | n$, $m = \Omega(\ell \log n)$ and $k = o(\sqrt{\ell})$. Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. For sufficiently large n and ℓ, there is an $n \overset{\varepsilon}{\mapsto} m$ f-RAC with PR and bias $\varepsilon \geq (1 - o_n(1))\,\mathrm{Stab}_q[f]$ with*

$$q \geq \sqrt{\frac{m}{n} - \frac{5\log_2(n/\ell)}{n/\ell}}. \tag{4.10}$$

Proof. Consider a code $C \subseteq \{-1,1\}^n$ such that, for every $x \in \{-1,1\}^n$, there is a $y \in C$ within Hamming distance $(1 - p - \frac{1}{n})n$, with $p > 1/2$ (the extra $1/n$ term will be used to counterbalance the decrease in probability from Newman's theorem). It is known [50] that there is such a code C of size

$$\log_2 |C| = (1 - H(p + 1/n))\, n + 2\log_2 n \leq (1 - H(p))\, n + 4\log_2 n. \tag{4.11}$$

Let $C(x)$ denote the closest codeword to x. Hence at least $(p + 1/n)n$ out of n bits of $C(x)$ are the same as x, and the probability over a uniformly random i that $x_i = C(x)_i$ is at least $p + 1/n$.

Let $\ell \in \mathbb{N}$ such that ℓ divides n. Our protocol involves breaking up $x \in \{-1,1\}^n$ into ℓ parts and encoding each part with the above code $C \subseteq \{-1,1\}^{n/\ell}$. Define the map

$$C^{(\ell)}(x) = C(x_1 \ldots x_{n/\ell}) C(x_{n/\ell+1} \ldots x_{2n/\ell}) \ldots C(x_{(\ell-1)n/\ell+1} \ldots x_n) \tag{4.12}$$

that applies C to the first ℓ bits of x, and to next ℓ bits of x and so on. Hence the probability that $x_i = C^{(\ell)}(x)_i$ over a uniformly random i is at least $p + \ell/n$. In order to consider this probability for *every* bit instead of just an average over all bits, we employ the following randomization process. Let $r \in \{-1,1\}^n$ and $\pi \in \mathbb{S}_n$, both taken uniformly at random. Given $x \in \{-1,1\}^n$, denote $\pi(x) = x_{\pi(1)}x_{\pi(2)}\dots x_{\pi(n)}$. We define the encoding $C^{(\ell)}_{\pi,r}(x) := \pi^{-1}(C^{(\ell)}(\pi(x \cdot r))) \cdot r$, where $x \cdot r$ denotes the bit-wise product of x and r. Let $\mathcal{E}_S$ be the event that all indices in $S \in \mathcal{S}^k_n$ are encoded in different codes C, i.e., in different blocks from $C^{(\ell)}$. There are ℓ blocks, each with n/ℓ elements. The probability that k specific elements fall into k different blocks is

$$\Pr_{S\sim\mathcal{S}^k_n}[\mathcal{E}_S] = \left(\frac{n}{\ell}\right)^k \frac{\binom{\ell}{k}}{\binom{n}{k}} = \prod_{j=1}^{k-1} \frac{1-\frac{j}{\ell}}{1-\frac{j}{n}} \geq \prod_{j=1}^{k-1}\left(1-\frac{j}{\ell}\right) \overset{(a)}{\geq} 1 - \sum_{j=1}^{k-1}\frac{j}{\ell} = 1 - \frac{k(k-1)}{2\ell}, \tag{4.13}$$

where inequality (a) can easily be proven by induction or the union bound.

We shall first present a protocol using shared randomness, and at the end we shall transform it into a protocol with private randomness by using Newman's theorem. The protocol is the following. Select $r \in \{-1,1\}^n$ and $\pi \in \mathbb{S}_n$ uniformly at random. Encode x as $C^{(\ell)}_{\pi,r}(x)$. To decode $f(x_S)$, first we check if all the indices of S were encoded into different blocks. If no, the value for $f(x_S)$ is drawn uniformly at random. If yes, just consider $C^{(\ell)}_{\pi,r}(x)_S$ and evaluate $f(C^{(\ell)}_{\pi,r}(x)_S)$. Conditioned on the event $\mathcal{E}_S$ happening, the probability that $x_{S_i} = C^{(\ell)}(x)_{S_i}$ is at least $p + \ell/n$ independently for all $i \in [k]$, meaning that

$$\Pr_{\pi,r}[f(x_S) = f(C^{(\ell)}_{\pi,r}(x)_S)|\mathcal{E}_S] \geq \Pr_{\substack{(x,y)\\ (q+\frac{2\ell}{n})\text{-correlated}}}[f(x) = f(y)] = \frac{1}{2} + \frac{1}{2}\operatorname{Stab}_{q+\frac{2\ell}{n}}[f], \tag{4.14}$$

where $q := 2p-1$, and the inequality follows from monotonicity of the noise stability of f. With these considerations, the success probability of the protocol is

$$\begin{aligned}
\Pr_{\pi,r}[f(x_S) = f(C^{(\ell)}_{\pi,r}(x)_S)] &> \frac{1}{2}\frac{k(k-1)}{2\ell} + \left(1 - \frac{k(k-1)}{2\ell}\right)\left(\frac{1}{2} + \frac{1}{2}\operatorname{Stab}_{q+\frac{2\ell}{n}}[f]\right) && (4.15)\\
&\geq \frac{1}{2} + \frac{1}{2}\left(1 - \frac{k(k-1)}{2\ell}\right)\left(\operatorname{Stab}_q[f] + \operatorname{Stab}_{\frac{2\ell}{n}}[f]\right) && (4.16)\\
&= \frac{1}{2} + \frac{1}{2}(1-o_n(1))\left(\operatorname{Stab}_q[f] + \operatorname{Stab}_{\frac{2\ell}{n}}[f]\right), && (4.17)
\end{aligned}$$

where we used that $k = o(\sqrt{\ell})$.

We now transform the shared randomness into private randomness. By Newman's theorem there is a small set of permutation-string pairs (note that Alice's input is size n bits and Bob's input is at most n bits) with size

$$t \leq \frac{4n}{(1-o_n(1))\operatorname{Stab}_{\frac{2\ell}{n}}[f]^2} \leq \frac{n^{2k+1}}{(1-o_n(1))2^{2k-2}\ell^{2k}} \tag{4.18}$$

(we have used that $\operatorname{Stab}_a[f] \geq a^k$) such that $f(x_S) = f(C^{(\ell)}_{\pi,r}(x)_S)$ continues to hold with probability at least $\frac{1}{2} + \frac{1}{2}(1-o_n(1))\operatorname{Stab}_q[f]$ if π, r are chosen uniformly at random from this set.

Hence the randomization can be encoded together with x at the cost of a small overhead. The final protocol chooses $j \in [t]$ uniformly at random, encodes x as $C^{(\ell)}_{\pi_j, r_j}(x)_S$ and then proceeds like the protocol with shared randomness. Fix $m = \log_2(t|C^{(\ell)}|)$. The result follows by using the first inequality from Theorem 4.15 to observe that

$$m \leq \big(1 - H(p)\big)n + 4\ell \log_2 \frac{n}{\ell} + \log_2 \frac{n^{2k+1}}{(1 - o_n(1))2^{2k-2}\ell^{2k}} \leq q^2 n + 4(1 + o_\ell(1))\ell \log_2 \frac{n}{\ell} \quad (4.19)$$

$$\implies 2p - 1 \geq \sqrt{\frac{m}{n} - \frac{4(1 + o_\ell(1)) \log_2 (n/\ell)}{n/\ell}}, \quad (4.20)$$

for sufficiently large n, where we used $k = o(\sqrt{\ell})$ again. ■

Remark. The parameter ℓ in Theorem 4.17 controls the number of encoding blocks in the protocol. By tweaking it, we can adjust the range of m and k, e.g. if $\ell = \Theta(\log n)$, then $m = \Omega(\log^2 n)$ and $k = o(\sqrt{\log n})$. If $\ell = \Theta(\sqrt{n})$, then $m = \Omega(\sqrt{n} \log n)$ and $k = o(n^{1/4})$.

In the protocol from Theorem 4.17 we broke the initial string into ℓ different blocks and used ℓ different copies of C. This was done in order to guarantee the independence of the $C(x)_{S_i}$'s and hence analyse the influence of the code C on the function f. Interestingly enough, for the special case of the function Parity this is not required and a single copy of C can be used.

Theorem 4.18. *Let* $\mathrm{XOR}_k : \{-1, 1\}^k \to \{-1, 1\}$ *be the Parity function and let* $m = \Omega(k \log n)$. *There is an* $n \overset{\varepsilon}{\mapsto} m$ XOR_k*-RAC with PR and bias*

$$\varepsilon \geq \frac{1}{\binom{n}{k}} \mathcal{K}_{k,n} \left(\frac{n}{2} - \frac{n}{2} \sqrt{\frac{m}{n} - \frac{7k \log_2 n}{n}} \right), \quad (4.21)$$

where $\mathcal{K}_{k,n}(x) = \sum_{j=0}^{k} (-1)^j \binom{x}{j} \binom{n-x}{k-j}$ *is the Krawtchouk polynomial.*

Proof. Consider the encoding $C_{\pi,r}(x) = \pi^{-1}(C(\pi(x \cdot r))) \cdot r$, where $C \subseteq \{-1, 1\}^n$ is the code described in Theorem 4.17. Let δn be the Hamming distance between x and $C(x)$, with $\delta \leq 1 - p - 1/n$ by the properties of C. Then

$$\Pr_{S \sim \mathcal{S}_n^k} \left[\prod_{i=1}^{k} x_{S_i} = \prod_{i=1}^{k} C(x)_{S_i} \right] = \sum_{\ell=0}^{\lfloor \frac{k}{2} \rfloor} \frac{\binom{(1-\delta)n}{k-2\ell} \binom{\delta n}{2\ell}}{\binom{n}{k}} = \frac{1}{2} + \frac{1}{2} \frac{\mathcal{K}_{k,n}(\delta n)}{\binom{n}{k}} \geq \frac{1}{2} + \frac{1}{2} \frac{\mathcal{K}_{k,n}((1-p)n)}{\binom{n}{k}} + \frac{1}{\binom{n}{k}}, \quad (4.22)$$

where we used $\sum_{\ell=0}^{k} \binom{(1-\delta)n}{k-\ell} \binom{\delta n}{\ell} = \binom{n}{k}$ on the second equality and $\mathcal{K}_{k,n}(\delta n) - \mathcal{K}_{k,n}(\delta n + 1) = 2$ on the final inequality, which can be obtained via the recurrence relation $\mathcal{K}_{k,n}(x) - \mathcal{K}_{k,n}(x-1) = \mathcal{K}_{k-1,n}(x) - \mathcal{K}_{k-1,n}(x-1)$ and $\mathcal{K}_{1,n}(x) = n - 2x$ (see e.g. [51]).

By Newman's theorem (Theorem 4.16) there is a small set of permutation-string pairs with size $t = n\binom{n}{k}^2$ such that $\prod_{i=1}^{k} x_{S_i} = \prod_{i=1}^{k} C(x)_{S_i}$ continues to hold with bias at least $\mathcal{K}_{k,n}((1-p)n)/\binom{n}{k}$ for *any* x and S if π, r are chosen uniformly at random from this set.

Our protocol is the following. Select $j \in [t]$ uniformly at random. Encode x as $C_{\pi_j,r_j}(x)$. To decode $\prod_{i=1}^k x_{S_i}$, just consider $C_{\pi_j,r_j}(x)_S$ and evaluate $\prod_{i=1}^k C_{\pi_j,r_j}(x)_{S_i}$. Now fix $m = \log_2(t|C|)$. By using the first inequality from Theorem 4.15 to observe that

$$m \le \big(1 - H(p)\big)n + 5\log_2 n + 2\log_2 \binom{n}{k} \implies 1 - p \le \frac{1}{2} - \frac{1}{2}\sqrt{\frac{m}{n} - \frac{7k\log_2 n}{n}}, \tag{4.23}$$

the result follows. ∎

Remark. Since $\mathcal{K}_{1,n}(x) = n - 2x$, we note that, for $k = 1$, Eq. (4.21) reduces to $\varepsilon \ge \sqrt{\frac{m}{n} - \frac{7\log_2 n}{n}}$, which is the result from Ambainis *et al.* [12] (see Theorem 4.2).

Remark. If $k = O(1)$, then the Krawtchouk polynomial has the asymptotic limit as $n \to \infty$ of $\mathcal{K}_{k,n}(x) \sim \frac{2^k n^k}{k!}\left(\frac{1}{2} - \frac{x}{n}\right)^k$ [63, Eq. (29)], thus the bias from Theorem 4.18 has the asymptotic limit of

$$\varepsilon \sim \frac{n^k}{k!\binom{n}{k}}\left(\frac{m}{n} - \frac{7k\log_2 n}{n}\right)^{k/2} \ge \left(\frac{m}{n} - \frac{7k\log_2 n}{n}\right)^{k/2}. \tag{4.24}$$

Note that this result is very similar to the one that would follow from Theorem 4.17, but slightly tighter (without the ℓ parameter and the multiplicative constant $1 - o_n(1)$).

4.2.2 f-RAC with SR

There is a lower limit of $m = \Omega(\log n)$ on the number of encoded bits in Theorem 4.17. It is possible to go below this limit by using SR: the blocks are now encoded via the $n \overset{\varepsilon}{\mapsto} 1$ RAC with SR from Theorem 4.1 instead of the code C.

Theorem 4.19. *Let $m|n$ and $k = o(\sqrt{m})$. Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. There is an $n \overset{\varepsilon}{\mapsto} m$ f-RAC with SR and bias $\varepsilon \ge (1 - o_n(1))\,\mathrm{Stab}_q[f]$ with*

$$q = \frac{2}{2^{n/m}}\binom{n/m - 1}{\lfloor \frac{n/m-1}{2} \rfloor} \ge \begin{cases} \sqrt{\frac{2m}{\pi n}} - O\left(\frac{1}{(n/m)^{3/2}}\right), \\ \sqrt{\frac{m}{2n}}. \end{cases} \tag{4.25}$$

Proof. Consider the RAC with SR from Theorem 4.1. Our protocol is the following. For the encoding of $x \in \{-1,1\}^n$, its n bits are randomly divided into m sets $T_1, \ldots, T_m$, each with n/m elements. Each set is encoded into 1 bit with the $n/m \overset{\epsilon}{\mapsto} 1$ RAC, and the encoded string is $E(x) \in \{-1,1\}^m$. For decoding, one checks with the help of SR if all the k indices in S were encoded into different sets. If no, the value for $f(x_S)$ is drawn uniformly at random. If yes, let $D_i : \{-1,1\} \to \{-1,1\}$ be the decoding function for set T_i, which corresponds to bit $E(x)_i$. Then $z_{\ell_i} := D_{\ell_i}(E(x)_{\ell_i})$ is the decoded x_{S_i}, where, for all $i \in [k]$, $\ell_i \in [m]$ is such that $S_i \in T_{\ell_i}$.[5] Write $z_T = z_{\ell_1} \ldots z_{\ell_k}$. We output $f(z_T)$ for $f(x_S)$.

[5] The decoding map of the RAC from Theorem 4.1 (see [11, Theorem 2]) is just the identity map, so $z_{\ell_i} = E(x)_{\ell_i}$.

Let $\mathcal{E}_S$ be the event that all indices in $S \in \mathcal{S}_n^k$ are encoded in different sets. Similarly to Eq. (4.13),

$$\Pr_{S\sim\mathcal{S}_n^k}[\mathcal{E}_S] \geq 1 - \frac{k(k-1)}{2m}. \tag{4.26}$$

The bias of correctly recovering any of the n/m encoded bits is $q = \frac{2}{2^{n/m}}\binom{n/m-1}{\lfloor\frac{n/m-1}{2}\rfloor}$ by Theorem 4.1. Conditioning on $\mathcal{E}_S$ happening, we see that x_S and y_T are q-correlated according to Definition 1.4. Therefore

$$\Pr_{T_1,\dots,T_m}[f(x_S) = f(z_T)|\mathcal{E}_S] = \Pr_{\substack{(x,y)\\ q\text{-correlated}}}[f(x) = f(y)] = \frac{1}{2} + \frac{1}{2}\operatorname{Stab}_q[f], \tag{4.27}$$

where we used an input randomization via SR. With these considerations, the success probability of the protocol is

$$\begin{aligned}\Pr_{T_1,\dots,T_m}[f(x_S) = f(z_T)] &\geq \frac{1}{2}\frac{k(k-1)}{2m} + \left(1 - \frac{k(k-1)}{2m}\right)\left(\frac{1}{2} + \frac{1}{2}\operatorname{Stab}_q[f]\right) &(4.28)\\ &= \frac{1}{2} + \frac{1}{2}\left(1 - \frac{k(k-1)}{2m}\right)\operatorname{Stab}_q[f] &(4.29)\\ &= \frac{1}{2} + \frac{1}{2}(1 - o_n(1))\operatorname{Stab}_q[f], &(4.30)\end{aligned}$$

using that $k = o(\sqrt{m})$, from where the result follows by noticing that

$$q = \frac{2}{2^{n/m}}\binom{n/m-1}{\lfloor\frac{n/m-1}{2}\rfloor} \geq \begin{cases}\sqrt{\frac{2m}{\pi n}} - O\left(\frac{1}{(n/m)^{3/2}}\right),\\ \sqrt{\frac{m}{2n}},\end{cases} \tag{4.31}$$

with $n! = \sqrt{2\pi n}(n/e)^n(1 + O(1/n))$ and $\binom{2n}{n} \geq 2^{2n}/(2\sqrt{n})$ [137, Chapter 10, Lemma 7]. ■

Remark. It is possible to use Newman's theorem in the above theorem in order to transform SR into PR, but then $\Omega(\log n)$ encoding bits would need to be used to encode the randomization procedure, thus leading to $m = \Omega(\log n)$. Moreover, the final f-RAC would have worse bias compared to the one from Theorem 4.17.

Remark. The requirement $m|n$ can be dropped by adding extra bits into $x \in \{-1, 1\}^n$ until $m|n'$, where n' is the final number of bits.

Remark. For $m = n/2$ or $m = n/3$, we have $\frac{2}{2^{n/m}}\binom{n/m-1}{\lfloor\frac{n/m-1}{2}\rfloor} = \frac{1}{2}$, so the resulting biases have $q_{m=n/2} = q_{m=n/3} = 1/2$. One can prove by induction that $\frac{2}{2^n}\binom{n-1}{\lfloor\frac{n-1}{2}\rfloor} \leq \frac{1}{2}$ for any $n \geq 2$.

4.2.3 f-QRAC

The same procedure from Theorem 4.19 holds for f-QRACs if we replace the $n/m \mapsto 1$ RAC from Theorem 4.1 with the $n/m \mapsto 1$ QRAC from Theorem 4.3 when encoding the sets $T_1, \dots, T_m$.

Theorem 4.20. *Let $m|n$ and $k = o(\sqrt{m})$. Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. There is an $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with SR and bias $\varepsilon \geq (1 - o_n(1)) \operatorname{Stab}_q[f]$ with*

$$q = \sqrt{\frac{8m}{3\pi n}} + O\left(\frac{1}{(n/m)^{3/2}}\right). \tag{4.32}$$

Proof. Replace the $n/m \mapsto 1$ RAC in the proof of Theorem 4.19 with the $n/m \overset{\epsilon}{\mapsto} 1$ QRAC from Theorem 4.3 with bias $\epsilon = \sqrt{\frac{8m}{3\pi n}} + O\left(\frac{1}{(n/m)^{3/2}}\right)$. ■

Remark. If $m = n/2$ or $m = n/3$, the usual (and optimal) $2 \overset{1/\sqrt{2}}{\longmapsto} 1$ QRAC or $3 \overset{1/\sqrt{3}}{\longmapsto} 1$ QRAC with PR can be used, respectively. The resulting biases have $q_{m=n/2} = 1/\sqrt{2}$ and $q_{m=n/3} = 1/\sqrt{3}$.

4.2.4 f-EARAC

The same protocol can also be used for f-EARACs, now with the $n/m \mapsto 1$ EARAC from Theorem 4.5.

Theorem 4.21. *Let $m|n$ and $k = o(\sqrt{m})$. Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. There is an $n \overset{\varepsilon}{\mapsto} m$ f-EARAC with bias $\varepsilon \geq (1 - o_n(1)) \operatorname{Stab}_q[f]$ and*

$$q = \sqrt{\frac{m}{n}}. \tag{4.33}$$

Proof. Replace the $n/m \mapsto 1$ RAC in the proof of Theorem 4.19 with the $n/m \overset{\epsilon}{\mapsto} 1$ EARAC from Theorem 4.5 with bias $\epsilon = \sqrt{m/n}$. ■

Remark. We could also define an entanglement-assisted f-QRAC (f-EAQRAC) similarly to Definition 4.10, i.e., as an f-QRAC with SR where both parties share an unlimited amount of entanglement. Due to super-dense coding and teleportation, an $n \overset{\varepsilon}{\mapsto} m$ f-EAQRAC is equivalent to an $n \overset{\varepsilon}{\mapsto} 2m$ f-EARAC, meaning that there is an $n \overset{\varepsilon}{\mapsto} m$ f-EAQRAC with $k = o(\sqrt{m})$ and bias $\varepsilon \geq (1 - o_n(1)) \operatorname{Stab}_q[f]$ with $q = \sqrt{\frac{2m}{n}}$.

If $f : \{-1,1\} \to \{-1,1\}$ is $f(x) = x$, i.e., when considering the usual $n \mapsto m$ EARAC, the above result tells us that the decoding probability is just

$$p = \frac{1}{2} + \frac{1}{2}\sqrt{\frac{m}{n}}. \tag{4.34}$$

Moreover, we note that the $n \mapsto m$ EARAC is formed by a grouping of $n/m \mapsto 1$ EARACs, such that the m outcomes, i.e., the bits of the encoding message $E(x) \in \{-1,1\}^m$, are all independent of each other. More precisely, for each $i \in [n]$, there is a unique $j \in [m]$ such that $\Pr[x_i|E(x)] = \Pr[x_i|E(x)_j]$. Under this assumption, we can prove that Eq. (4.34) is optimal by

the optimality of its parts. Consider breaking the initial string $x \in \{-1,1\}^n$ into m blocks, the ith block containing r_i elements. Then the success probability of the final $n \mapsto m$ EARAC is

$$\sum_{i=1}^{m} \frac{r_i}{n}\left(\frac{1}{2}+\frac{1}{2\sqrt{r_i}}\right) = \frac{1}{2}+\frac{1}{2n}\sum_{i=1}^{m}\sqrt{r_i}, \tag{4.35}$$

which is maximized by taking $r_i = n/m$ for all $i \in [m]$. Since the $n/m \mapsto 1$ EARACs are optimal by Theorem 4.5, then so is Eq. (4.34) (under the assumption that the $n \mapsto m$ EARAC is formed by m independent EARACs on 1 encoded bit).

Theorem 1.25 combined with Theorems 4.17, 4.19, 4.20, 4.21 leads to the following corollary.

Corollary 4.22. *Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function with* pure high degree *$\hbar = \min\{|S| : \widehat{f}(S) \neq 0\}$. Let $W^j[f] = \sum_{\substack{S\subseteq[k]\\|S|=j}} \widehat{f}(S)^2$.*

(a) Let $\ell \in \mathbb{N}$, $m = \Omega(\ell \log n)$ and $k = o(\sqrt{\ell})$. There is an $n \overset{\varepsilon}{\mapsto} m$ f-RAC with PR and bias $\varepsilon \geq (1-o_n(1))W^{\hbar}[f]\left(\frac{m}{n} - \frac{5\log_2(n/\ell)}{n/\ell}\right)^{\hbar/2}$.

(b) Let $k = o(\sqrt{m})$. There is an $n \overset{\varepsilon}{\mapsto} m$ f-RAC with SR and bias $\varepsilon \geq (1-o_n(1))W^{\hbar}[f]\left(\frac{m}{2n}\right)^{\hbar/2}$.

(c) Let $k = o(\sqrt{m})$. There is an $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with SR and bias $\varepsilon \geq (1-o_n(1))W^{\hbar}[f]\left(\frac{8m}{3\pi n}\right)^{\hbar/2}$.

(d) Let $k = o(\sqrt{m})$. There is an $n \overset{\varepsilon}{\mapsto} m$ f-EARAC with bias $\varepsilon \geq (1-o_n(1))W^{\hbar}[f]\left(\frac{m}{n}\right)^{\hbar/2}$.

4.2.5 f-PRRAC

We now present a protocol for the f-PRRAC, based on reducing the problem to the standard random access code setting, and then using a protocol defined in [165]. This protocol was used to show the violation of information causality by means of a pyramid of Popescu-Rohrlich boxes and nesting van Dam's protocol [55], which allows us to decode the value of $f(x_S)$ for any $S \in \mathcal{S}_n^k$ with just one encoded bit. This procedure of pyramiding and nesting was also used in the context of EARACs in [166, 189] under the name of concatenation.

Theorem 4.23. *Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. For any $n \in \mathbb{N}$, there is an $n \overset{1}{\mapsto} 1$ f-PRRAC.*

Proof. To ease the notation, we shall use $\{0,1\}$ instead of $\{-1,1\}$ during the proof. We shall also name the encoding and decoding parties Alice and Bob, respectively, and refer to a Popescu-Rohrlich box as PR-box. Let $t := |\mathcal{S}_n^k| = k!\binom{n}{k}$ [6] and define the string $a \in \{0,1\}^t$ as $a_S := f(x_S)$, where $\mathcal{S}_n^k$ is arranged in lexicographic order.[7] Bob is interested in bit a_S, whose index position can be described by a t-bit string $b = \sum_{i=0}^{t-1} b_i 2^i$, i.e., the considered function

[6] If f is symmetric, t can be decreased to $\binom{n}{k}$ by ignoring all the redundant permutations of the k-sets.
[7] Here we use a_S to denote a single bit of a, whereas x_S denotes a subsequence of x.

is $g_t(a,b) := a_b$. The remainder of the argument is the same as the protocol of [165], but we include the details for completeness. For $t = 1$ we have that

$$g_t((a_0,a_1),b_0) = a_0 \oplus b_0(a_0 \oplus a_1). \tag{4.36}$$

Alice inputs $a_0 \oplus a_1$ into a PR-box, while Bob inputs b_0. Alice obtains the output A and sends the message $y = a_0 \oplus A$ to Bob, who can obtain $y \oplus B = a_b$ using his output B, since, by the PR-box property, $A \oplus B = b_0(a_0 \oplus a_1)$.

For $t > 1$, write $a = a'a''$, where $a' = a_0 \dots a_{t/2-1} \in \{0,1\}^{t/2}$ and $a'' = a_{t/2} \dots a_{t-1} \in \{0,1\}^{t/2}$. Then one can show that

$$g_t(a,b) = g_{t-1}(a',b') \oplus b_{t-1}\left(g_{t-1}(a',b') \oplus g_{t-1}(a'',b')\right), \tag{4.37}$$

where $b' = b_0 \dots b_{t-2} \in \{0,1\}^{t-1}$. Therefore we can construct a recursive protocol in t, which will encompass all values of n. The protocol uses a pyramid of $2^t - 1$ Popescu-Rohrlich boxes placed on t levels. The case $t = 1$ was explained above. For $t > 1$, Alice and Bob use the protocol on inputs (a',b') and (a'',b'), which involves $2^{t/2} - 1$ PR-boxes in each one. Alice's outputs of each protocol are y' and y'', which she inputs into the last PR-box, similarly to the case $t = 1$, as $y' \oplus y''$, while Bob inputs b_{t-1}. Given Alice's final output A, she sends $y = y' \oplus A$ to Bob, who uses his output B_{t-1} to obtain $y' \oplus b_{t-1}(y' \oplus y'')$. If $b_{t-1} = 0$, he gets y', otherwise, if $b_{t-1} = 1$, he gets y''. With these, he can recursively go up the pyramid based on the protocol for $t-1$ bits, which tells him which boxes to read. Looking at the binary decomposition of b, Bob goes $(t-r)$ times to the left bit, and r times to the right bit, where $r = \sum_{i=0}^{t-1} b_i$. His final output will be $y \oplus B_0 \oplus \dots \oplus B_{t-1}$, where B_j is the output for the PR-box that Bob uses at level j. Bob will only need the outputs of t PR-boxes, while Alice uses $2^t - 1$ PR-boxes in total. ■

4.3 Bias Upper Bounds

In order to prove an upper bound on the bias of any $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with SR, we use the following equivalent version of Definition 4.9, which comes from input randomization, i.e., from considering the average success probability over the inputs, and from the following fact.

Fact 4.24 ([98])**.** *Let ρ be an unknown state picked from the set $\{\rho_0,\rho_1\}$ with probability p and $1-p$, respectively. The optimal success probability of predicting which state it is by a POVM is*

$$\frac{1}{2} + \frac{1}{2}\|p\rho_0 - (1-p)\rho_1\|_{\mathrm{tr}}. \tag{4.38}$$

Definition 4.25. Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. An $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with SR is a map $\rho : \{-1,1\}^n \to \mathbb{C}^{2^m \times 2^m}$ that assigns an m-qubit density matrix $\rho(x)$ to every $x \in \{-1,1\}^n$ and has the property that

$$\mathop{\mathbb{E}}_{S \sim \mathcal{S}_n^k}\left[\frac{1}{2^n}\left\|\sum_{x\in\{-1,1\}^n} f(x_S)\rho(x)\right\|_{\mathrm{tr}}\right] \geq \varepsilon. \tag{4.39}$$

Theorem 4.26. *Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. For any $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with SR and $k = o(n)$ the following holds: for any $\eta > 2\ln 2$ there is a constant C_η such that*

$$\varepsilon \leq C_\eta \sum_{\ell=0}^{k} L_{1,\ell}(f) \left(\frac{\eta m}{n}\right)^{\ell/2}, \tag{4.40}$$

where $L_{1,\ell}(f) = \sum_{\substack{T\subseteq[k]\\|T|=\ell}} \left|\widehat{f}(T)\right|$ is the 1-norm of the ℓ-th level of the Fourier transform of f.

Proof. We start by writing the following.

$$\frac{1}{2^n}\left\|\sum_{x\in\{-1,1\}^n} f(x_S)\rho(x)\right\|_{\mathrm{tr}} = \frac{1}{2^n}\left\|\sum_{V\subseteq[n]}\sum_{T\subseteq[k]} \widehat{f}(T)\widehat{\rho}(V)\sum_{x\in\{-1,1\}^n}\chi_V(x)\chi_T(x_S)\right\|_{\mathrm{tr}} \tag{4.41}$$

$$= \left\|\sum_{T\subseteq[k]} \widehat{f}(T)\widehat{\rho}(S_T)\right\|_{\mathrm{tr}} \tag{4.42}$$

$$\leq \sum_{T\subseteq[k]} \left|\widehat{f}(T)\right| \|\widehat{\rho}(S_T)\|_{\mathrm{tr}}, \tag{4.43}$$

where $S_T = \{S_i \mid i \in T\}$. Then

$$\varepsilon \leq \sum_{\ell=0}^{k}\sum_{\substack{T\subseteq[k]\\|T|=\ell}} \left|\widehat{f}(T)\right| \mathop{\mathbb{E}}_{S\sim\mathcal{S}_n^k}\left[\|\widehat{\rho}(S_T)\|_{\mathrm{tr}}\right], \tag{4.44}$$

but, for a given $T \subseteq [k]$ with $|T| = \ell$,

$$\mathop{\mathbb{E}}_{S\sim\mathcal{S}_n^k}\left[\|\widehat{\rho}(S_T)\|_{\mathrm{tr}}\right] = \frac{1}{k!\binom{n}{k}}\sum_{S\in\mathcal{S}_n^k}\|\widehat{\rho}(S_T)\|_{\mathrm{tr}} = \frac{\ell!(k-\ell)!}{k!\binom{n}{k}}\binom{n-\ell}{k-\ell}\sum_{S\in\binom{[n]}{\ell}}\|\widehat{\rho}(S)\|_{\mathrm{tr}} = \mathop{\mathbb{E}}_{S\sim\binom{[n]}{\ell}}\left[\|\widehat{\rho}(S)\|_{\mathrm{tr}}\right], \tag{4.45}$$

and thus, using Jensen's inequality,

$$\varepsilon \leq \sum_{\ell=0}^{k}\sum_{\substack{T\subseteq[k]\\|T|=\ell}} \left|\widehat{f}(T)\right| \mathop{\mathbb{E}}_{S\sim\binom{[n]}{\ell}}\left[\|\widehat{\rho}(S)\|_{\mathrm{tr}}\right] \leq \sum_{\ell=0}^{k} L_{1,\ell}(f)\sqrt{\mathop{\mathbb{E}}_{S\sim\binom{[n]}{\ell}}\left[\|\widehat{\rho}(S)\|_{\mathrm{tr}}^2\right]}. \tag{4.46}$$

We now use Theorem 1.31 with $\delta = \frac{\ell}{(2\ln 2)m}$, taking only the sum on S with $|S| = \ell$,

$$\mathop{\mathbb{E}}_{S\sim\binom{[n]}{\ell}}\left[\|\widehat{\rho}(S)\|_{\mathrm{tr}}^2\right] \leq 2^{2\delta m}\delta^{-\ell}\binom{n}{\ell}^{-1} = \left(\frac{(2e\ln 2)m}{\ell}\right)^{\ell}\binom{n}{\ell}^{-1}, \tag{4.47}$$

to finally obtain

$$\varepsilon \leq \sum_{\ell=0}^{k}\binom{n}{\ell}^{-1/2} L_{1,\ell}(f)\left(\frac{(2e\ln 2)m}{\ell}\right)^{\ell/2}. \tag{4.48}$$

From here we use Stirling's approximation $n! = \Theta(\sqrt{n}(n/e)^n)$ to obtain

$$\binom{n}{\ell} = \frac{n!}{\ell!(n-\ell)!} = \Theta\left(\sqrt{\frac{n}{\ell(n-\ell)}}\left(\frac{n}{\ell}\right)^{\ell}\left(1+\frac{\ell}{n-\ell}\right)^{n-\ell}\right). \tag{4.49}$$

We use the fact that for large enough n/ℓ we have $(1+\ell/(n-\ell))^{(n-\ell)/\ell} > (2e\ln 2)/\eta$, where $\eta > 2\ln 2$, and that the factor $\sqrt{n/\ell(n-\ell)} \geq \sqrt{1/k}$ can be absorbed by this approximation. Then there is a constant C_η such that Eq. (4.40) holds. ■

Many different bounds can be obtained from the above theorem, some with a clearer interpretation.

Corollary 4.27. *Let $f : \{-1,1\}^k \to \{-1,1\}$ be a Boolean function. Let $r \in [0,1]$. For any $n \overset{\varepsilon}{\mapsto} m$ f-QRAC with SR and $k = o(n)$ the following holds: for any $\eta > 2\ln 2$ there is a constant C_η such that*

$$\varepsilon \leq \begin{cases} C_\eta \sqrt{\sum_{S\in\operatorname{supp}(\widehat{f})} q^{2(1-r)|S|}}\sqrt{\operatorname{Stab}_{q^{2r}}[f]}, & \text{(4.50a)} \\ C_\eta \hat{\|} f \hat{\|}_1 \left(\frac{\eta m}{n}\right)^{\hbar/2}, & \text{(4.50b)} \\ C_\eta 2^{\deg(f)-1}\operatorname{Stab}_q[f], & \text{(4.50c)} \end{cases}$$

where $q = \sqrt{\frac{\eta m}{n}}$, $\operatorname{supp}(\widehat{f}) = \{S \subseteq [k] : \widehat{f}(S) \neq 0\}$ is the support of f, $\hbar = \min\{|S| : \widehat{f}(S) \neq 0\}$ is the pure high degree of f, $\deg(f) = \max\{|S| : \widehat{f}(S) \neq 0\}$ is the degree of f and $\hat{\|} f \hat{\|}_1 = \sum_{S\subseteq[k]} |\widehat{f}(S)|$ is the Fourier 1-norm of f.

Proof. From Theorem 4.26 we know that for any $\eta > 2\ln 2$ there is a constant C_η such that

$$\varepsilon \leq C_\eta \sum_{\ell=0}^{k} L_{1,\ell}(f)\left(\frac{\eta m}{n}\right)^{\ell/2}. \tag{4.51}$$

There are a few ways to bound the above quantity. We start by proving Eq. (4.50a). Define $g : \{-1,1\}^k \to \mathbb{R}$, $g = \sum_{S\in\operatorname{supp}(\widehat{f})} \operatorname{sgn}(\widehat{f}(S))\chi_S$. Let T_q be the noise operator with parameter $q = \sqrt{\frac{\eta m}{n}}$. Let $r, s \in [0,1]$ be such that $r + s = 1$. By Cauchy-Schwarz,

$$\left(\sum_{S\subseteq[k]} q^{|S|}|\widehat{f}(S)|\right)^2 = |\langle \mathrm{T}_{q^r} f, \mathrm{T}_{q^s} g\rangle|^2 \leq \langle \mathrm{T}_{q^r} f, \mathrm{T}_{q^r} f\rangle\langle \mathrm{T}_{q^s} h, \mathrm{T}_{q^s} g\rangle = \operatorname{Stab}_{q^{2r}}[f] \sum_{S\in\operatorname{supp}(\widehat{f})} q^{2s|S|}. \tag{4.52}$$

By plugging Eq. (4.52) into Eq. (4.51), Eq. (4.50a) follows.

Other ways of bounding are

$$\varepsilon \leq C_\eta \sum_{\ell=0}^{k} L_{1,\ell}(f)\left(\frac{\eta m}{n}\right)^{\ell/2} \leq C_\eta \hat{\|} f \hat{\|}_1 \left(\frac{\eta m}{n}\right)^{\hbar/2} \tag{4.53}$$

or

$$\varepsilon \leq C_\eta \sum_{\ell=0}^{k} L_{1,\ell}(f) \left(\frac{\eta m}{n}\right)^{\ell/2} \leq C_\eta \sum_{\ell=0}^{k} 2^{\deg(f)-1} W^\ell[f] \left(\frac{\eta m}{n}\right)^{\ell/2} = C_\eta 2^{\deg(f)-1} \operatorname{Stab}_q[f], \quad (4.54)$$

where we used that f's Fourier spectrum is $2^{1-\deg(f)}$-granular in Eq. (4.54), i.e., $\widehat{f}(S)$ is an integer multiple of $2^{1-\deg(f)}$ for all $S \subseteq [k]$ [157, Exercise 1.11]. ■

Corollary 4.27 helps with the comparison between the bias upper bound and the bias lower bounds for f-RAC and f-QRAC (Theorems 4.17, 4.19 and 4.20). By taking $\deg(f)$ as constant, Eq. (4.50c) matches the bias lower bounds in terms of the noise stability up to an overall multiplicative constant and up to the multiplicative constant $\sqrt{\eta}$ in the parameter q in $\operatorname{Stab}_q[f]$. Another comparison is between Eq. (4.50b) and Corollary 4.22 in terms of the pure high degree of f. Again both bounds match up to a global multiplicative constant and up to the constant η. We conjecture that the constant η can be dropped from all these bounds with a better analysis.

4.4 Conclusions

In this chapter we proposed a simple generalization of the concept of random access to recovering the value of a given Boolean function on any subset of fixed size of the initial bits. This generalization was made assuming different resources as encoding maps, i.e., encoding the initial string into bits or qubits, and different auxiliary resources, e.g. private and shared randomness, shared entanglement and Popescu-Rohrlich boxes. Given the lower bounds from our protocols, it seems reasonable to assume that the bias $\operatorname{Stab}_q[f]$ with $q = \sqrt{\frac{m}{n}}$ is, if not optimal, at least close to optimal. The case with the weakest resources, the $n \mapsto m$ f-RAC with PR, already achieves such bias up to an additive term $O((\log n/\ell)/(n/\ell))$ in the parameter q. For more general values of $m = O(\log n)$, the use of quantum resources progressively improves q: from $q \approx \sqrt{\frac{2m}{\pi n}}$ using encoding bits and SR to $q \geq \sqrt{\frac{8m}{3\pi n}}$ using encoding qubits and SR and finally to $q = \sqrt{\frac{m}{n}}$ using encoding bits and shared entanglement. Such an improvement offered by quantum resources is relatively modest, specially when compared to stronger-than-quantum resources like Popescu-Rohrlich boxes, which allows the recovery of $f(x_S)$ with certainty for any S.

On the other hand, the techniques from Fourier analysis lead to bias upper bounds that match our bias lower bounds up to a global multiplicative constant and a factor $\sqrt{\eta} \approx \sqrt{2 \ln 2}$ in the parameter q. We conjecture that such upper bounds can be improved and the factor η dropped. Moreover, the upper bounds apply only to f-QRACs with SR, therefore not including f-EARACs. The understanding of EARACs is still limited, and even though we obtained an upper bound by making an independence assumption, a general upper bound for the case $m > 1$ is yet unknown. Actually, a general tight upper bound (better than Nayak's bound) for the case $m > 1$ is unknown even for usual QRACs (the $m = 2$ case was recently solved by Mančinska and Storgaard [138]).

Part II

Chapter 5

Quantum Error Correction: the Toric Code

In this chapter we review the toric code introduced by Kitaev [118, 117], which is an example of a *stabilizer code*, i.e., a quantum error correcting code based on the *stabilizer formalism* invented by Gottesman [86]. Other examples of stabilizer codes are Shor's nine-qubit code [179], Steane's code [185], the five-qubit code [29, 131] and the CSS (Calderbank-Shor-Steane) codes [45, 186]. An introduction to the stabilizer formalism is the chapter 10 from Nielsen and Chuang's textbook [155] and Gottesman's paper [88]. In regard to the toric code, see Dennis, Kitaev, Landahl and Preskill's seminal paper [58] and Dan Browne's lecture notes [36].

5.1 The Stabilizer Formalism

The stabilizer formalism studies error correcting codes from the perspective of *operators*, and not quantum states. Instead of explicitly writing down the quantum states belonging to the *codespace* of the code, i.e., the subspace of the Hilbert space used to define the code, we can analyse the operators 'behind' the codespace (its *stabilizers* as defined below). From a physics point of view, the stabilizer formalism is equivalent to working in the Heisenberg picture [87].

The stabilizer formalism draws its power from clever use of group theory. Here, and when studying the toric code, the group of interest is the *Pauli group* $\mathbf{P}_n$. On one qubit, the Pauli group $\mathbf{P}_1$ consists of all *Pauli operators*, together with overall factors ± 1 and $\pm i$,

$$\mathbf{P}_1 := \{\pm\mathbb{I}, \pm i\mathbb{I}, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}, \tag{5.1}$$

where the Pauli operators/matrices are defined as usual:

$$\mathbb{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \; Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \; Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{5.2}$$

The multiplicative constants ± 1 and $\pm i$ guarantee the closure of $\mathbf{P}_1$ under matrix multiplication. The Pauli group $\mathbf{P}_n$ on n-qubits is defined as $\mathbf{P}_n := \mathbf{P}_1^{\otimes n}$, i.e., it consists of all n-fold tensor products of single-qubit Pauli operators from $\mathbf{P}_1$, so any element $P \in \mathbf{P}_n$ can be expressed as $P = A_1 \otimes A_2 \otimes \cdots \otimes A_n$, where $A_i \in \mathbf{P}_1$ for all $i \in [n]$.

Given an Abelian subgroup $\mathbf{S}$ of $\mathbf{P}_n$, we define $V_{\mathbf{S}}$ as the vector subspace of all n-qubit states that are simultaneous $+1$ eigenstates of every element of $\mathbf{S}$, i.e.,

$$V_{\mathbf{S}} := \{|\psi\rangle \in \mathscr{H} : S|\psi\rangle = |\psi\rangle \ \forall S \in \mathbf{S}\}. \tag{5.3}$$

We say that an operator $S \in \mathbf{S}$ *stabilizes* a quantum state $|\psi\rangle$ if $S|\psi\rangle = |\psi\rangle$, i.e., if S leaves the quantum state unchanged. Therefore $V_{\mathbf{S}}$ is the *vector space stabilized* by $\mathbf{S}$, and $\mathbf{S}$ is said to *stabilize* $V_{\mathbf{S}}$. For a subgroup $\mathbf{S}$ of the Pauli group to stabilize a nontrivial vector space, two conditions are necessary and sufficient: (a) the elements of $\mathbf{S}$ commute and (b) $-\mathbb{I} \notin \mathbf{S}$ (and $\pm i\mathbb{I} \notin \mathbf{S}$ by extension). That these conditions are necessary is straightforward to see, otherwise one would arrive at a contradiction of the kind $|\psi\rangle = -|\psi\rangle$. See [155] for a proof of their sufficiency.

A (sub)group $\mathbf{S}$ can be represented by its *generators*. A set of elements $S_1, \ldots, S_k \in \mathbf{S}$ generates the (sub)group $\mathbf{S}$ if any element $S \in \mathbf{S}$ can be written as $S = \prod_{j=1}^{k} S_j^{a_j}$ where $a_j = \{0, 1\}$. This is denoted by $\mathbf{S} = \langle S_1, \ldots, S_k \rangle$. Moreover, we say that a set of generators is *independent* if the only solution to $\prod_{j=1}^{k} S_j^{a_j} = \mathbb{I}$ is $a_j = 0$ for all $j \in [k]$. If $\mathbf{S} = \langle S_1, \ldots, S_k \rangle$ is generated by k independent operators, then the solution to all equations $S_j|\psi\rangle = |\psi\rangle$ determines k degrees of freedom of the state $|\psi\rangle$. Since it also has dimension 2^n, the remaining $n - k$ degrees of freedom make up the codespace, i.e., the state space spanned by logical states. Logical states and, by extension, logical qubits, are specific states made up of physical bits/qubits that abstractly play the role of $|0\rangle$ and $|1\rangle$, and which are determined by the underlying classical/quantum code. We have the following proposition.

Proposition 5.1 ([155, Proposition 10.5])**.** *Let $\mathbf{S} = \langle S_1, \ldots, S_k \rangle$ be generated by k independent and commuting elements from $\mathbf{P}_n$, and such that $-\mathbb{I} \notin \mathbf{S}$. Then the number m of encoded logical qubits by the stabilizer code is $m = n - k$.*

Together with the stabilizers, the codespace defined by $\mathbf{S}$ also possesses logical operators that act on logical qubits. By denoting $|\bar{0}\rangle_i$ and $|\bar{1}\rangle_i$ the logical states of the i-th qubit, the logical operators $\bar{X}_i$ and $\bar{Z}_i$ are defined by the relations

$$\bar{X}_i|\bar{0}\rangle_i = |\bar{1}\rangle_i, \quad \bar{X}_i|\bar{1}\rangle_i = |\bar{0}\rangle_i, \tag{5.4a}$$

$$\bar{Z}_i|\bar{0}\rangle_i = |\bar{0}\rangle_i, \quad \bar{Z}_i|\bar{1}\rangle_i = -|\bar{1}\rangle_i. \tag{5.4b}$$

In order to be valid logical operators, $\bar{X}_i$ and $\bar{Z}_i$ need to commute with all the stabilizers, but not be contained in the stabilizer group. Moreover, they must also anticommute with each other.

If an error $E \in \mathbf{P}_n$ happens to a codestate $|\psi_c\rangle$, it can be identified by measuring all stabilizer generators. Since the stabilizers commute with logical operators, their measurement does not disturb the encoded information. In an error-free state, all the stabilizer measurements would return a '+1' outcome, thus any '−1' outcome signals the presence of an error. More specifically, given that an error operator E and a stabilizer S_i belong to $\mathbf{P}_n$, E and S_i must either commute or anticommute. If they commute,

$$S_i E|\psi_c\rangle = ES_i|\psi_c\rangle = E|\psi_c\rangle, \tag{5.5}$$

so the state $E|\psi_c\rangle$ is still an '+1' eigenstate of S_i. If they anticommute,

$$S_i E|\psi_c\rangle = -ES_i|\psi_c\rangle = -E|\psi_c\rangle, \tag{5.6}$$

and $E|\psi_c\rangle$ is now an '−1' eigenstate of S_i, which indicates an error. The sequence of all outcomes from the stabilizer generator measurements form a *syndrome* $\mathbf{s}$. This can be viewed as partitioning the Hilbert space into different subspaces depending on the eigenvalues of the stabilizer generators, i.e., $\mathcal{H} = \bigoplus_{\mathbf{s}} \mathcal{H}_{\mathbf{s}}$, where $\mathbf{s} = (s_1, \ldots, s_k)$ with $s_i = \pm 1$ for all $i \in [k]$ is the syndrome vector. The codespace of the stabilizer group is just $\mathcal{H}_{(+1,+1,\ldots,+1)}$. Measuring all the stabilizer generators returns a syndrome vector $\mathbf{s}$, which is used to determine an *error correction* C aimed at restoring the state to the codespace, i.e., $CE|\psi_c\rangle = S|\psi_c\rangle$, where S is an element of the stabilizer group. The process, or algorithm, in determining an error correction from the syndrome is called a *decoder*. Devising good decoders is not a simple task, and we shall review a few for the toric code later in this chapter, and propose new ones in Chapter 6.

5.2 The Toric Code

The *toric code* is best understood via a spatial arrangement of qubits in an $L \times L$ square lattice with periodic boundary conditions. The periodic boundary conditions are such that the right-most edge matches with the left-most edge and the upper edge matches with the lower edge, as if the lattice wrapped around a torus (see Fig. 5.1). This lattice is made up of *edges*, *vertices* (points where edges meet) and *plaquettes* (individual squares enclosed by edges). Every *edge* of the lattice has a qubit (these correspond to circles in Fig. 5.1).

The toric code is an example of a stabilizer code, and as such can be best understood via its stabilizer group, which is generated by two types of operators (also called check operators): *plaquette* operators and *star* operators (Fig. 5.2). A star operator $S_v^\star$ is associated with vertex v of the toric code and consists of a tensor product of Pauli X operators acting on the 4 qubits adjacent to vertex v, and identity operators acting on the rest of the qubits, i.e.,

$$S_v^\star = \bigotimes_{\ell \ni v} X_\ell. \tag{5.7}$$

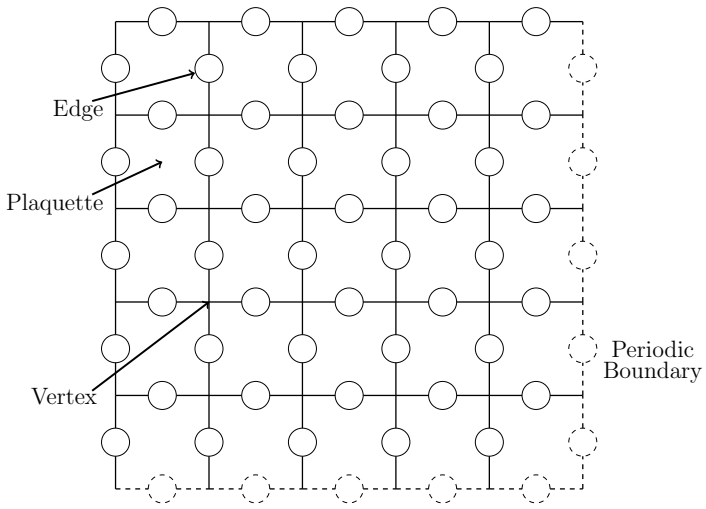


Figure 5.1: The toric code is defined on an $L \times L$ square lattice with periodic boundary conditions, and it consists of *edges*, *vertices* and *plaquettes*. Each circle placed at an edge represents a qubit.

A plaquette operator $S_P^{\square}$, on the other hand, is associated with plaquette P and consists of a tensor product of Pauli Z operators acting on the 4 qubits adjacent to plaquette P, and identity operators acting on the remaining qubits, i.e.,

$$S_P^{\square} = \bigotimes_{\ell \in P} Z_\ell. \tag{5.8}$$

It is not hard to see that the stabilizer operators all commute, as required. Plaquette operators obviously commute with plaquette operators, and star operators with star operators. Moreover, plaquette operators also commute with star operators since they either act on disjoint qubits, or on sets with two intersecting qubits. In the second case, the two minus signs from the anticommuting operators at the intersecting qubits cancel each other.

In an $L \times L$ lattice with periodic boundaries, there are $2L^2$ edges (qubits), L^2 plaquettes and L^2 vertices. However, every star or plaquette operator can be expressed as a product of the other $L^2 - 1$ such operators, since $\prod_v S_v^{\star} = \prod_P S_P^{\square} = \mathbb{I}$. Thus there are $2(L^2 - 1)$ independent check operators. From Proposition 5.1 it follows that a toric code encodes 2 logical qubits.

A cycle formed out of edges is said to be *trivial* if its interior can be 'tiled' by plaquettes, i.e., if it is the boundary of a set of plaquettes. Therefore a product of plaquette operators has a boundary which is a trivial cycle (the same can be said about star operators, but in the *dual lattice*). However, a cycle could also be nontrivial, i.e., not the boundary of anything. Such nontrivial cycles loop around the torus and are not contained in the stabilizer group. Associated with the two fundamental nontrivial cycles (horizontal and vertical loops around the torus) are the logical operators $(\bar{X}_1, \bar{Z}_1)$ and $(\bar{X}_2, \bar{Z}_2)$. In other words, the logical operators are formed from X or Z operators acting on chains of qubits that loop around the torus. One can see from Fig. 5.2 that the logical operators commute with all the stabilizers, and themselves are

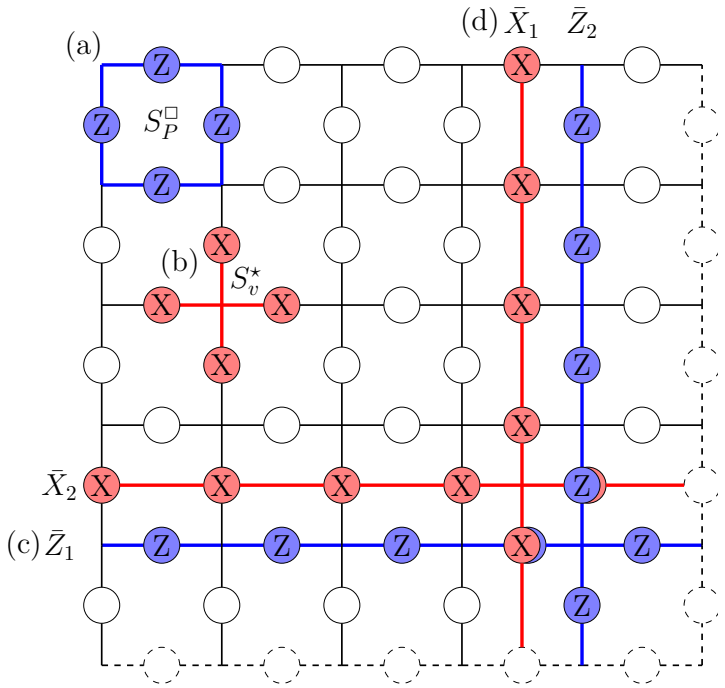


Figure 5.2: The stabilizer group of the toric code is made up of (a) plaquette operators, defined on each plaquette of the code, and (b) star operators, defined on each vertex of the code. The logical operators (c) $\bar{Z}$ and (d) $\bar{X}$ are associated with nontrivial cycles of Z and X operators, respectively.

not part of the stabilizer group. Moreover, one can also see that the logical operators $\bar{X}_1, \bar{Z}_1$ anticommute, since they overlap on exactly one qubit (and similarly to $\bar{X}_2, \bar{Z}_2$).

5.3 Syndrome

One of the great advantages of the toric code is that its stabilizer operators are simple. Each one involves only just four qubits that are close to one another, meaning that the operators are local. Therefore it is easy to measure all plaquette and star operators in order to obtain the syndrome, as previously mentioned in Section 5.1. If there are no errors in the code, then all check operators will return an $+1$ outcome. On the other hand, any single Pauli error will lead to two -1 measurement outcomes, as shown in Fig. 5.3. An X error will flip the value of the two plaquette operators that contained the affected qubit, while an Z error will flip the value of the two star operators. X and Z errors are referred to as *bit-flip* and *phase* errors, respectively.

It is helpful to view the -1 measurement outcomes as particles called *anyons*. The anyons from star operators are placed at the vertices of the lattice, while anyons from plaquette operators are placed at the plaquettes of the lattice. A single error creates a pair of anyons (Fig. 5.3), while a chain of errors has one anyon at each of its ends (Fig. 5.4a), which means that extending the error chain propagates the pair along the lattice. As an example consider a chain of Z errors

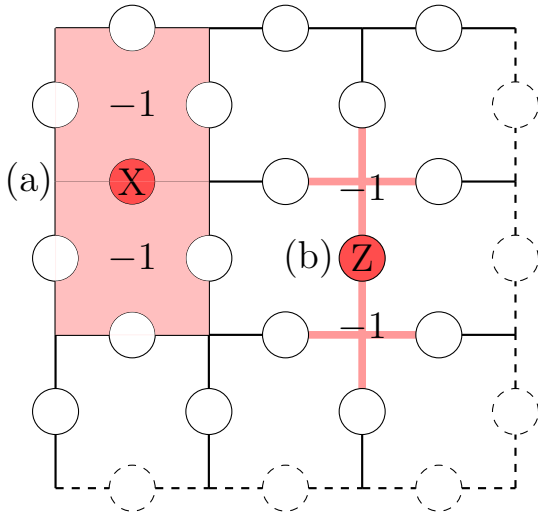
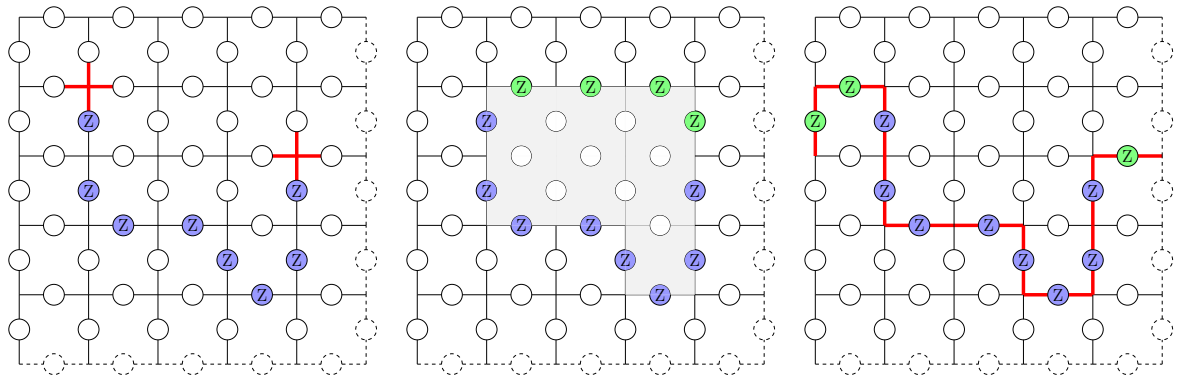


Figure 5.3: A single qubit error creates a pair of -1 stabilizer measurement outcomes, or anyons. (a) An X error is detected by the two adjacent plaquette operators, while (b) an Z error is detected by the two adjacent star operators.

from Fig. 5.4a. The anyons are present at the two ends of the chain (its boundary), where the star operators have their outcomes flipped. The same is true for X errors. A chain of X errors flips the two plaquette operators at its two ends. We note that a general error configuration will be a disjoint union of continuous error chains. From now on, by error chain we shall mean either a single continuous error chain or a disjoint union of them.

A correction operator can be obtained from the syndrome. Unfortunately, there is not a one-to-one correspondence between syndrome and error chains. Any error chain with boundary at the anyons generates the same syndrome. In other words, any two Pauli error operators E and E' will lead to the same syndrome if $E = E'S$, where S is any operator which commutes with the stabilizer group. As an example, if we again consider the error chain from Fig. 5.4a and use it as a correction operator, this would successfully erase all errors and recover the code back to the codespace. Now suppose we choose another chain E' with the same boundary as E for our correction operator, as shown in Fig. 5.4b. In this case the errors do not cancel out, but form a trivial cycle. Nonetheless, such trivial cycle belongs to the stabilizer group and therefore the code is corrected back to the codespace. Another scenario would be choosing a correction operator E' with the same boundary as E such that EE' now forms a nontrivial cycle, as shown in Fig. 5.4c. Even though stabilizer measurements cannot detect the presence of errors after applying E', the code does not belong to the codespace anymore, since a nontrivial cycle is equivalent to applying a logical operator. We say in this case that there is a *logical error*.

In summary, any correction operator consists of pairing and joining the anyons together. The correction is successful if all cycles at the end of the correction are trivial, otherwise there is a logical error. The problem is deciding on the *right pairing*, which is the task of the decoder.



(a) A blue Z error chain creates a pair of anyons.

(b) A green Z chain is applied, generating a trivial cycle.

(c) A green Z chain is applied, generating a nontrivial cycle.

Figure 5.4: (a) The blue Z error chain creates a pair of anyons, marked in red, at its boundary. In order to correct it, a green chain with the same syndrome as the blue error chain is applied to code. (b) The code is successfully corrected if the resulting Z chain is a trivial cycle, i.e., can be tiled with plaquette operators (shown in grey). (c) If the cycle is nontrivial (shown in red), there is a logical error and the correction failed.

5.4 Faulty Measurements

Until now we have assumed that all stabilizer measurements are performed perfectly. Unfortunately, this will be far from the truth in real experiments. Faulty measurements will have some probability of returning a wrong result, e.g. the measurement is flipped with some probability. In order to gather information about the faulty measurements, we repeat the syndrome measurement a sufficient number of times. This procedure, however, opens the possibility of new errors happening in between stabilizer measurements. The net result is an "error history", which generates a syndrome. One can see from Fig. 5.5 that a physical error at time t creates a pair of -1 measurement outcomes that repeat through time, while a measurement error creates a single -1 measurement outcome in a specific point in time. We now treat *locations where one stabilizer measurement differs from its value in the previous round* as an anyon. Therefore a measurement error creates a pair of anyons separated in *time* (vertically), while a physical error creates a pair of anyons separated in *space* (horizontally).

Such syndrome, in turn, can be best visualized as a 3-dimensional array of $+1$ and -1 outcomes, with the third dimension representing an integer-valued *time*. A physical error that occurs at time t is associated with a horizontal (spacelike) edge from time layer t, while the outcome of a star operator at vertex v is placed at the vertical (timelike) edge connecting vertex v from times t and $t+1$. The same applies to plaquette operators, but at the dual lattice. From this syndrome history represented as a 3-dimensional array we can identify several chains (set of edges). A *syndrome chain* is the set of all vertical edges with nontrivial syndrome, i.e., with -1 stabilizer measurement outcomes (depicted in blue in Fig. 5.6). An *error chain* is

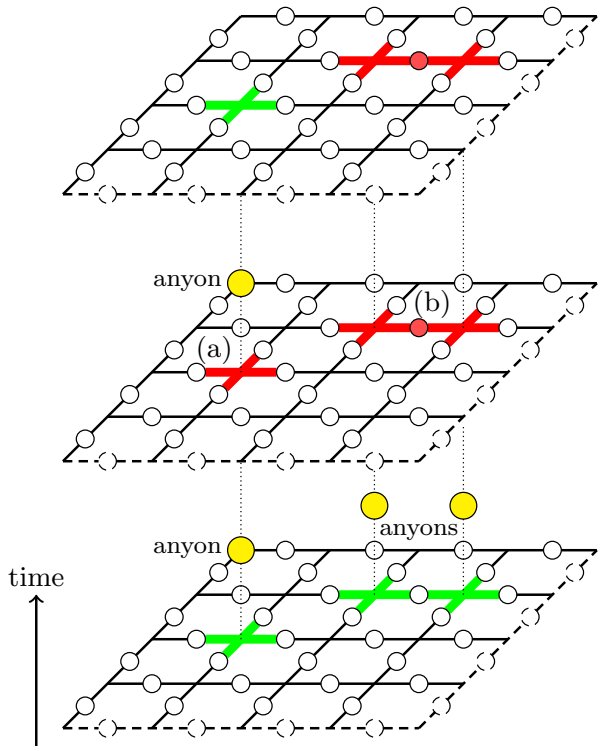


Figure 5.5: The toric code depicted through time, where rounds of stabilizer measurements are performed periodically. Star operators with $+1$ outcome are marked in green, while star operators with -1 outcome are marked in red (just a few measurements are highlighted). Yellow circles represent anyons that signal locations where a stabilizer measurement changes from one round to the next. (a) A measurement error produces a -1 outcome and a pair of anyons separated in time. (b) A physical error affecting the qubit marked in red produces a pair of -1 outcomes that propagate through time and a pair of anyons separated in space.

formed from all horizontal edges containing a physical error and all vertical edges where a measurement error occurred (depicted in red and green, respectively, in Fig. 5.6). Similarly to the perfect measurement scenario (see Fig. 5.4), the anyons are located at the *boundary* of the syndrome chain. Moreover, both syndrome and error chains share the same boundary, which is an indication of the error chain degeneracy.

Once the anyons are obtained from the 3D syndrome, we can treat them exactly as in the perfect measurement (2D) case: a correction operator is obtained from pairing and joining the anyons together. This correction operator is *projected* onto the final time layer and applied to the code. One can see from Fig. 5.6 that the physical errors present at the final time layer are obtained from projecting the error chain: only those horizontal links contained an odd number of time layers survive. As long as the projected error chain and the projected correction operator form only trivial cycles, the correction is successful.

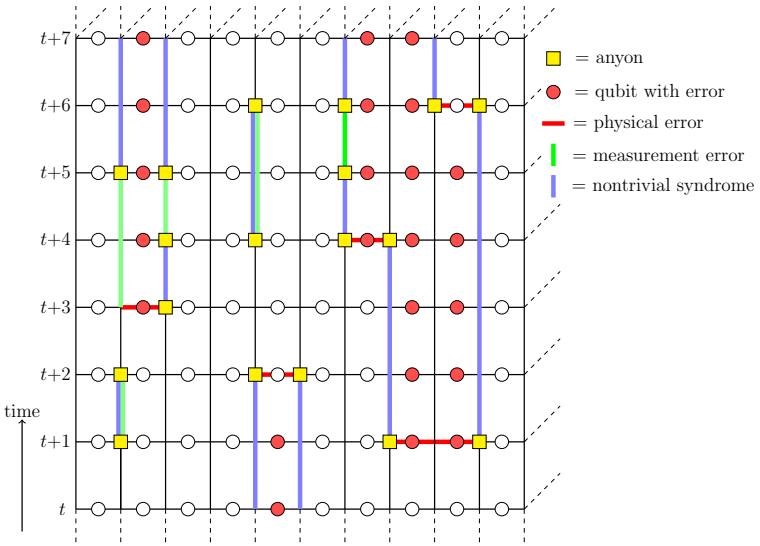


Figure 5.6: The history of the error syndrome in the toric code represented as a 3-dimensional array. Physical errors happen at integer-valued times $t, t+1, \ldots$, and are represented by a horizontal red line. Qubits can acquire a physical error (marked in red) or 'lose' an error, since two phase/bit-flips cancel each other. Stabilizer measurements are performed in between the integer-valued time layers, and nontrivial syndrome (-1 outcomes) are represented by a vertical blue line. Measurement errors are represented by a vertical green line. Anyons, which are the locations where a stabilizer measurement differs from its previous value, are shown as yellow squares. Similarly to Fig. 5.5, physical errors create pairs of anyons separated in space, while measurement errors create pairs of anyons separated in time. Note that the anyons are located at the boundary of the syndrome chain (blue lines), and that the error chain, i.e., physical (red lines) plus measurement (green lines) errors, has the same boundary as the syndrome chain.

5.5 Decoding

The choice of the correction operator, i.e., which anyons should be paired up, to return the state back to the codespace is the task of the *decoder*. Since there are many error chains that could have created the same syndrome, the decoder can never perfectly correct the code. Its object is to rather reduce the chances of a logical error.

5.5.1 Threshold

The toric code displays a *threshold behaviour* against errors [5, 124]. This means that, if the physical error rate p is below some value p_{th}, called the threshold and which depends on the underlying error model and decoder, the failure probability of the recovery procedure vanishes exponentially as the linear size L of the lattice increases to infinity. More formally, given an error chain E, let $E' = ED$ be a hypothetical error chain identified as a correction operator, where D is a cycle. Let $\Pr[ED|E]$ be the normalized conditional probability for error chains $E' = ED$ that have the same boundary as E. Then, the physical error rate per qubit is below

the threshold if and only if

$$\lim_{L\to\infty} \sum_{E} \Pr[E] \sum_{D \text{ nontrivial}} \Pr[ED|E] = 0. \tag{5.9}$$

The above equation says that correction operators that leave nontrivial error cycles on the code have probability zero as the lattice size goes to infinity, meaning that a sufficiently large code succeeds in correcting the errors with high probability.

The threshold value depends on the error model and the details of the decoder. In order to benchmark codes, it is normally assumed that errors are independent and identically distributed (i.i.d.), an assumption that we will make in Chapter 6. We shall briefly revise two main decoders.

5.5.2 Optimal Decoder

The optimal decoder identifies the correction operator least likely to induce a logical error by analysing all possible error chains that could have led to the observed syndrome. Let E be the error chain affecting the code and S the observed syndrome chain. As previously seen, S and E differ by a cycle C, i.e., $E = SC$. From the syndrome chain the decoder needs to guess what the error chain actually was, say $E' = SC'$. As long as $EE' = CC'$ is homologically trivial (a trivial cycle), the correction is successful. Here we define a *homology class* as the set of chains whose pairwise combination is homologically trivial. In other words, we say that C and C' belong to the same homology class if CC' is homologically trivial. The toric code possess four homology classes: one associated with any trivial cycle, one associated with each of the two fundamental nontrivial cycles around the torus, and one associated with the combination of these two fundamental nontrivial cycles (i.e., a cycle that loops around the torus in both the horizontal and vertical directions). Therefore the correction procedure is reduced to the identification of the homology class of C. A homology class h in the toric code has probability

$$\Pr[h|S] = \frac{\sum_{C'\in h} \Pr[SC']}{\sum_{C'} \Pr[SC']}. \tag{5.10}$$

By identifying the most likely homology class h, any cycle C' can be drawn from h and a correction operator $E' = SC'$ applied onto the code. Unfortunately, calculating the probability of each homology class is computationally demanding, as one needs to consider an exponential number of cycles. Nonetheless, there are more efficient ways to tackle the decoding problem, as we shall see below.

For the scenario of perfect measurements and i.i.d. Pauli errors on the qubits, the optimal value of the threshold for the toric code was analytically estimated to be at least 3.7% [102, 58] and numerically measured to be 10.9% [101, 142]. In the case of i.i.d. Pauli errors and faulty measurements, with the measurement error probability equal to the physical error probability, the optimal threshold was analytically estimated to be at least 1.1% [58] and numerically measured to be 3.3% [160].

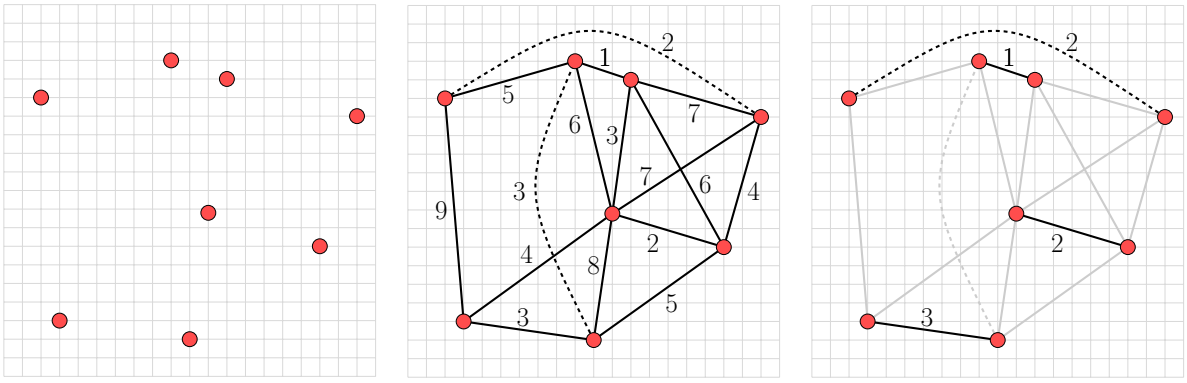


Figure 5.7: The MWPM decoder on the toric code. (a) Once the anyons are located, (b) a complete connected graph G is created by placing each anyon at its vertices. By associating a weight $\ln\left(\frac{1-p_e}{p_e}\right)$ to each edge of the torus, the weight of an edge of the graph G is the distance between the two anyons on the torus. Such distances can be calculated via Dijkstra's algorithm. In the specific case of i.i.d. errors, the weights of G are the shortest possible Manhattan distance between anyons on the torus. For clarity we omit edges with high weights from the figure. Edges that loop around the torus are shown as dashed lines. (c) Once the graph G is formed, the selected pairing is the matching with minimum additive weight.

5.5.3 Minimum Weight Perfect Matching Decoder

In order to get around the exponential-time computation of the optimal decoder, many efficient (polynomial time in L) decoding algorithms that find an approximate solution at the cost of reducing the threshold were proposed [202, 193, 106, 99, 66, 13, 195]. One of the most famous decoders is the Minimum Weight Perfect Matching (MWPM) decoder, based on an algorithm by Jack Edmonds from the 1960s [67] and used by Wang, Harrington and Preskill [193] to estimate the threshold of the toric code. The idea of the MWPM decoder is to find the *most likely* error chain that could have generated the syndrome, instead of finding the best correction operator.

To be more specifically, consider a toric code, either with perfect (2D lattice) or faulty (3D lattice) measurements, where all the errors are independent, but not necessarily identical. This means that to each edge e of the lattice is associated an error probability p_e. Therefore, the probability of an error chain E given the syndrome S is

$$\Pr[E|S] = \prod_{e\in E} p_e \prod_{e\notin E}(1-p_e) = c_0 \prod_{e\in E} \frac{p_e}{1-p_e}, \tag{5.11}$$

where $c_0 = \prod_{\forall e}(1-p_e)$ is a constant. By taking the logarithm, we see that

$$\max_E \ln \Pr[E|S] = \ln c_0 - \min_E \sum_{e\in E} \ln\left(\frac{1-p_e}{p_e}\right), \tag{5.12}$$

i.e., the most likely error chain is the one that minimizes $\sum_{e\in E} \ln\left(\frac{1-p_e}{p_e}\right)$. This minimisation can be accomplished in two steps: (1) determine the distance between every pair of anyons and

(2) find the error chain (pairing) that minimizes the sum of such distances. In order to determine the distance between two anyons (the minimum probability of any error chain connecting both anyons), i.e.,

$$\min_{F} \sum_{e \in F} \ln\left(\frac{1-p_e}{p_e}\right), \tag{5.13}$$

where F is a chain connecting both anyons, Dijkstra's algorithm [61] can be used (see Appendix 6.A for details on Dijkstra's algorithm). If all the errors are identical ($p_e = p$ for all e), then this distance is simply the *Manhattan distance* between the two anyons, i.e., the sum of their horizontal and vertical separation (and time separation in case of faulty measurements).[1]

Regarding the second step, the syndrome is mapped into a graph matching problem, which is explained in Fig. 5.7. All the anyons are seen as vertices of a complete graph G, and the edge weight between two anyons is the distance between them calculated in step one. Once the graph G is obtained, the pairing that produces the most likely error chain is the minimum weight perfect matching of G, which is obtained in polynomial time via, e.g. Edmonds' blossom algorithm [67] or Micali and Vazirani algorithm [143].

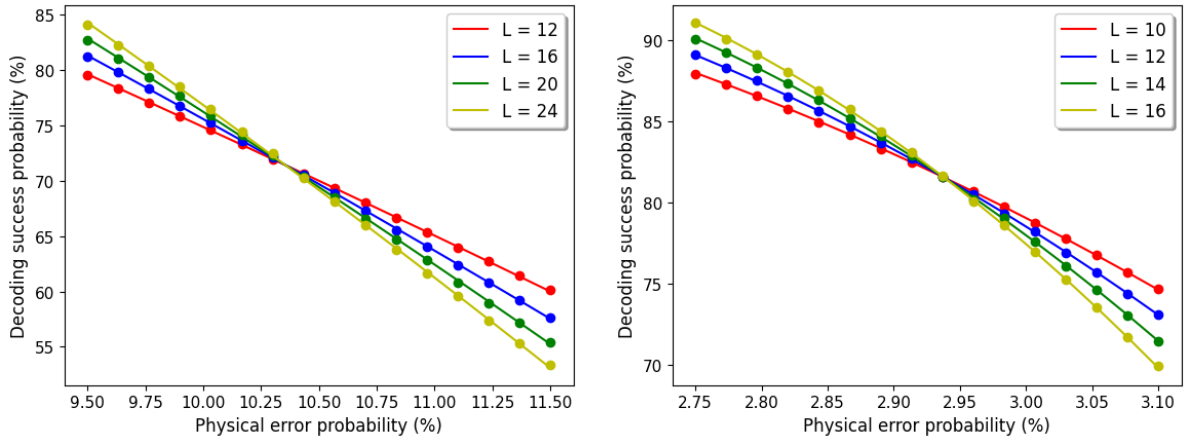
In Fig. 5.8 we show the results of our simulations of the MWPM decoder on the toric code under the i.i.d. error model and both perfect and faulty measurements. The plots show the success probability of the decoder, i.e., the probability that there is no logical error after applying the correction, as a function of the physical error p for several lattice sizes L. More specifically, Fig. 5.8a is obtained under perfect measurement. Every qubit in an $L \times L$ lattice initially suffers an error with probability p, and a single stabilizer measurement round is performed. The MWPM decoder is applied and we check if the code was successfully corrected. By repeating the procedure a sufficient number of times, the success probability of the MWPM decoder is calculated for the pair of parameters (L, p). Fig. 5.8b is similarly obtained under faulty measurements. We perform $2L$ measurement rounds, and in between each of them all the qubits suffer an error with probability p. Moreover, the outcome of a stabilizer measurement is flipped with probability $q = p$. The success probability of the MWPM decoder is again estimated by repeating the procedure a sufficient number of times.

The threshold can be identified from Figs. 5.8a and 5.8b as the crossing point of the curves for different L's. According to [193], the decoding success probability near the threshold is a function of $x = (p - p_{th})L^{1/\nu_0}$, where p_{th} is the threshold value and ν_0 is a parameter. Therefore, following [193], we estimate the threshold p_{th} via the quadratic approximation around $x = 0$:

$$P_{success} = A + B(p - p_{th})L^{1/\nu_0} + C(p - p_{th})^2 L^{2/\nu_0}, \tag{5.14}$$

where A, B, C, ν_0, p_{th} are fitting parameters, and $P_{success}$ is the decoding success probability. We obtain $p_{th}^{perfect} = 10.35\% \pm 0.01\%$ under perfect measurements and $p_{th}^{faulty} = 2.938\% \pm 0.001\%$ under faulty measurements. Both values agree with past results [193, 160, 188, 154, 95].

[1] Up to the multiplicative constant $\ln\left(\frac{1-p}{p}\right)$.



(a) Perfect measurements.

(b) Faulty measurements.

Figure 5.8: The decoding success probability of the MWPM decoder under i.i.d. Z errors for different lattice sizes L. (a) Stabilizer measurements are performed perfectly. The threshold value obtained is $p_{th}^{perfect} = 10.35\% \pm 0.01\%$. (b) Stabilizer measurements are faulty with error probability equal to the physical error. The number of measurement rounds equals $2L$. The threshold value obtained is $p_{th}^{faulty} = 2.938\% \pm 0.001\%$. Each simulation was repeated 10^6 times.

Chapter 6

Decoding Probabilistic Syndrome Measurements

It is usually assumed that stabilizer measurements can be made deterministically, i.e., on demand (not necessarily error-free), which is a reasonable assumption for many systems such as some experiments involving entangling gates between trapped ions [17]. Commonly, however, this assumption does not apply. In some systems, parity checks are inherently probabilistic, as is the case when they depend on ancillary states from non-deterministic entanglement generation or distillation procedures. This is a common issue for modular quantum-computing architectures [144, 154, 116]. In other systems, parity checks are subject to measurement erasure — i.e., the stabilizer measurements can always be attempted, but measurement outcomes are not always returned. For example, this applies to linear-optical quantum computing [123] using single-photon detectors, as they are subject to the effects of optical loss [85, 147, 146]. For either of these scenarios, it is natural to assume that parity checks are attempted over many rounds of repeated measurement, until stabilizer measurements are successfully recorded. For all decoders presented up to now, it has in fact been assumed that parity measurements can always be obtained at once — a situation we call *synchronous* stabilizer measurements.

6.1 Our Results

In this work we study a model of asynchronous parity check measurement on the toric code. In this model the stabilizer measurements are attempted at discrete times and each attempt provides a parity outcome with probability s, called the *synchronicity* parameter. We push this to the limit where parity checks are performed continuously, i.e., $s \to 0$. For the i.i.d. error model from Section 5.5.3 and a minimum weight perfect matching (MWPM) decoder [68, 125] the toric code exhibits a threshold of 2.93% when parity checks are entirely synchronous (see Figure 5.8a). We show that without modifying the decoder the threshold can still be maintained

at 1.178% as the asynchronicity increases, but that by appropriately modifying the decoder we can improve the threshold to 1.688% in the completely continuous regime. In Figure 6.1 we depict the performance of our main decoders with respect to the synchronicity.

A second aim of this chapter is to analyse the impact of *degeneracy* on decoders in the presence of asynchronism. It is known that the performance of the MWPM is close to optimal in many different cases [58, 193], and can be improved by taking advantage of degenerate minimum weight matchings [66, 184], which typically apply to large numbers of possible error configurations. This degeneracy behaves similarly to an entropy term in a quasi-free energy and has been used to close the gap between minimum-weight perfect matching and optimal methods [54], as well as to compare different variants of the surface code with a comparable number of qubits [30]. We provide numerical evidence that accounting for high-order degeneracy leads to little improvement on the threshold in regimes with high asynchronism: only an increase in threshold from 1.688% to 1.699% could be achieved.

In more details, the presence of asynchronism erases information in the 'history' of stabilizer measurement outcomes. In other words, it erases edges in the syndrome graph (see Figure 5.6). The net result is a time separation between measurement outcomes, and, consequently, the appearance of time blocks of erased parity checks, called *parity blocks*. Anyons are no longer well defined in time in such scenario. A simple approach is to apply the usual MWPM decoder to the erased syndrome graph, named *simple syndrome graph*, by identifying the parity blocks themselves with anyons. We name such decoder the *Unweighted Simple* (US) decoder, and its threshold monotonically decreases with the synchronicity until 1.178% at $s = 0$. A second simple approach is to place the anyons in the middle of the parity blocks and reduce the problem back to a cubic syndrome graph as in a full synchronous ($s = 1$) regime with i.i.d. error model. The resulting decoder is the *Average Position* (AP) decoder. At low values of s, it performs better than the Unweighted Simple decoder (for reasons we currently do not fully understand), achieving a threshold of 1.323% at $s = 0$.

We then propose a series of gradual improvements to this simple approach. The first is to contract the erased edges of the simple syndrome graph into multi-edges following the method described by Stace and Barrett [184]. This removes all edges marked as erased measurements from the simple syndrome graph and leads to the *contracted syndrome graph*. We then define the *Unweighted Contracted* (UC) decoder as a MWPM decoder applied to the contracted syndrome graph. The multi-edges obtained by contracting edges in the simple syndrome graph naturally account for multiple physical error configurations that connect a pair of syndromes, thus being a source of degeneracy and greatly improving the threshold. The Unweighted Contracted decoder maintains a threshold of 1.513% in the limit of continuous measurements.

The second improvement, introduced upon edge contraction, is to *weight* the edges of the contracted syndrome graph. The Unweighted Contracted decoder does not take into consideration non-identical error probabilities that arise from contracting the simple syndrome

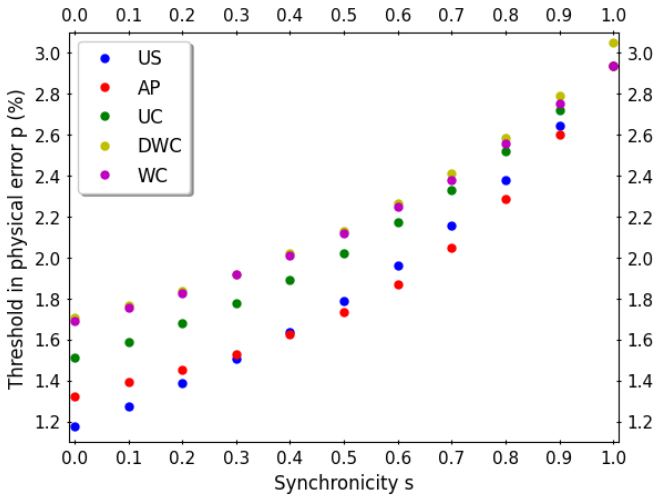


Figure 6.1: Threshold dependence of Unweighted Simple (US), Unweighted Contracted (UC), Weighted Contracted (WC), Degenerate Weighted Contracted (DWC) and Average Position (AP) decoders with synchronicity s.

graph. A decoder should prioritize a multi-edge formed of many different edges from the simple syndrome graph than a multi-edge made up of just a few. By weighting the multi-edges accordingly, we introduce the *Weighted Contracted* (WC) decoder, and another boost to the threshold can be obtained, e.g. 1.688% in the limit $s \to 0$.

The third and final improvement is to consider high-order degeneracy terms. The Weighted Contracted decoder is just a MWPM decoder after edge contraction and after weighting the resulting edges accordingly, thus it considers the most likely error chain that could have caused the observed syndrome. There is no need to stop there, though, and less likely error chains can be taken into consideration, which is referred to as high-order degeneracy terms. We can group the error chains by length, and by considering the set of shortest and second shortest error chains between a given pair of anyons, we define the *Degenerate Weighted Contracted* (DWC) decoder. However, the resulting improvement on the threshold is very small, and a value of 1.699% is achieved at the continuous measurement regime. We argue, and provide numerical evidence, that the presence of asynchronism increases the predominance, i.e., the relative probability, of the most likely error chain over all the others. This is in contrast to the full synchronous regime with i.i.d. error model, where normally there are a variety of different error chains with the same probability. Therefore, ignoring error chains other than the most likely one does not affect much the performance of the decoder.

The chapter is structured as follows. Section 6.2 introduces the model of asynchronism. Section 6.3 explains the formulation of the decoding problem and defines several decoders. Section 6.4 gives details on simulations. We benchmark the decoders' performance in Section 6.5.

6.2 Asynchronism in the toric code

6.2.1 Asynchronous Stabilizer Measurement

We introduce a model of asynchronous stabilizer measurement. This model is designed to isolate the effects of measurement asynchronism whilst leaving all other features of the system the same. But it is worth highlighting that there are many things about this model that would change depending on the physical system. In this model, which corresponds to an $L \times L$ lattice of qubits subject to repeated measurements:

1. Attempted stabilizer measurements provide definite ± 1 parity outcomes with probability s, which is referred to as the *synchronicity* parameter. Otherwise, with probability $1 - s$, no outcome is obtained, which is marked as a '0' outcome, i.e., erased.

2. Rounds of stabilizer measurements occur on a timescale so that a parity outcome of a stabilizer operator is obtained at an average rate of 1 per unit time (independently of the value of s). In other words, all stabilizers are measured once per unit time on average.

3. Qubit errors occur at a rate of p per unit time, simply referred to as the *physical error*. For simplicity only phase-flip errors and X-type parity checks are considered. By symmetry the performance will be the same for bit-flip errors, which use Z-type parity checks.

4. Parity outcomes are subject to measurement error with probability q, where the outcome value is flipped. It will be assumed that $q = p$.

The behaviour of the system with s is the main question of this chapter, which can be probed by fixing the rate of physical and measurement errors. We consider three distinct regimes with respect to the parameter s, which are illustrated in Figure 6.2:

1. **Synchronous measurement ($s = 1$).** This corresponds to the i.i.d. error model from Section 5.5.3 with fully synchronous parity checks.

2. **Discrete asynchronous measurement ($0 < s < 1$).** Measurements are performed in discrete rounds, but are not deterministic and occur with probability s. Measurement rounds are performed at a rate $\Delta t = 1/s$ such that the overall rate of successful stabilizer measurement remains at 1 per unit time.

3. **Continuous measurement ($s = 0$).** Measurements are not performed in rounds, but are received continuously at a rate 1. Similarly, Pauli errors are treated as continuous. The times of the measurements and Pauli errors are modelled as arising from a Poisson distribution.

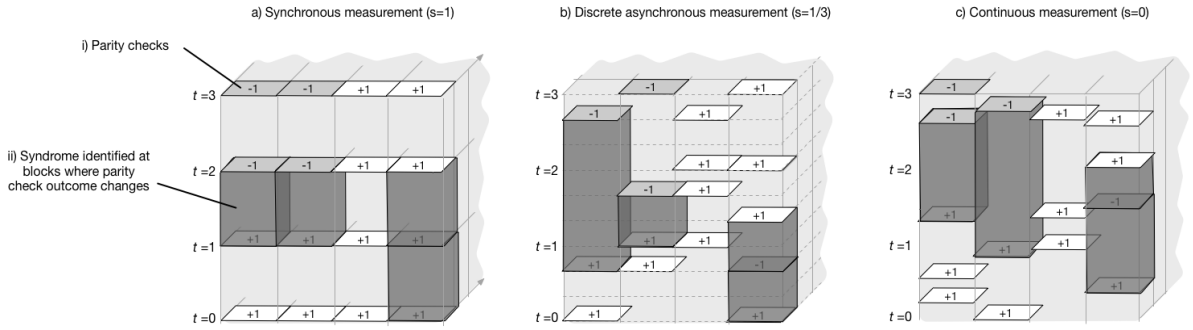


Figure 6.2: Illustration of three regimes of asynchronous stabilizer measurement in the square surface code. (a) Fully synchronous ($s = 1$). Repeated measurement on the code created a 3-dimensional block of parity check outcomes over time. Outcomes are measured deterministically in layers at discrete time intervals. An anyon (a violated syndrome bit) is identified when a parity check measurement changes from one round to the next (dark grey blocks). A physical error will result in two anyons separated in space. A measurement error will result in two anyons separated in time. (b) Discrete asynchronous measurement ($0 < s < 1$). Measurements are performed in discrete rounds, but an outcome is only returned with probability s, where in the figure $s = \frac{1}{3}$. Physical or measurement errors result in a pair of anyons, but these are now identified in intervals of varying size as indicated in the figure. (c) Continuous asynchronous ($s = 0$): parity check measurements can happen at any time, at a rate 1 per unit time.

One can see from Figure 6.2 the effect of the probabilistic nature of parity checks. Successful stabilizer measurements are separated in time, thus creating a block-like structure. Every stabilizer operator X_v has an ordered list of measurement times for successful parity checks $\mathcal{T}_v = (t_1^v, t_2^v, \dots)$, where $t_1^v < t_2^v < \dots$. Two consecutive measurement times define a *parity block*. More specifically, the i-th parity block associated with $v \in V$ is defined by the pair of time coordinates (t_{i-1}^v, t_i^v), and a stabilizer operator X_v has $|\mathcal{T}_v| - 1$ associated parity blocks. If the measurement outcomes differ from each other at consecutive times t_{i-1}^v and t_i^v, then we refer to this block as an *anyon block*. In the fully synchronous regime ($s = 1$), two consecutive measurements with differing outcomes led to an anyon well defined in time. On the other hand, for $s < 1$, such anyons (now anyon blocks) are spread over time.

6.2.2 Constructing the decoding problem

To analyze fault tolerance in this system we first want to formulate the error model and structure of the code as a *syndrome graph*. In the syndrome graph vertices represent fault tolerant parity checks and edges represent the potential errors in the system (see Figure 5.6). This representation is the most useful way to analyze the performance of decoding algorithms as it fully describes the system, capturing both space and time behavior.

Each edge in the syndrome graph is assigned a bit that indicates whether or not an error has occurred. Vertices are assigned a parity value which is computed as the parity of the values of all edges incident to that vertex. If there are no errors all vertices will have an even parity. If an

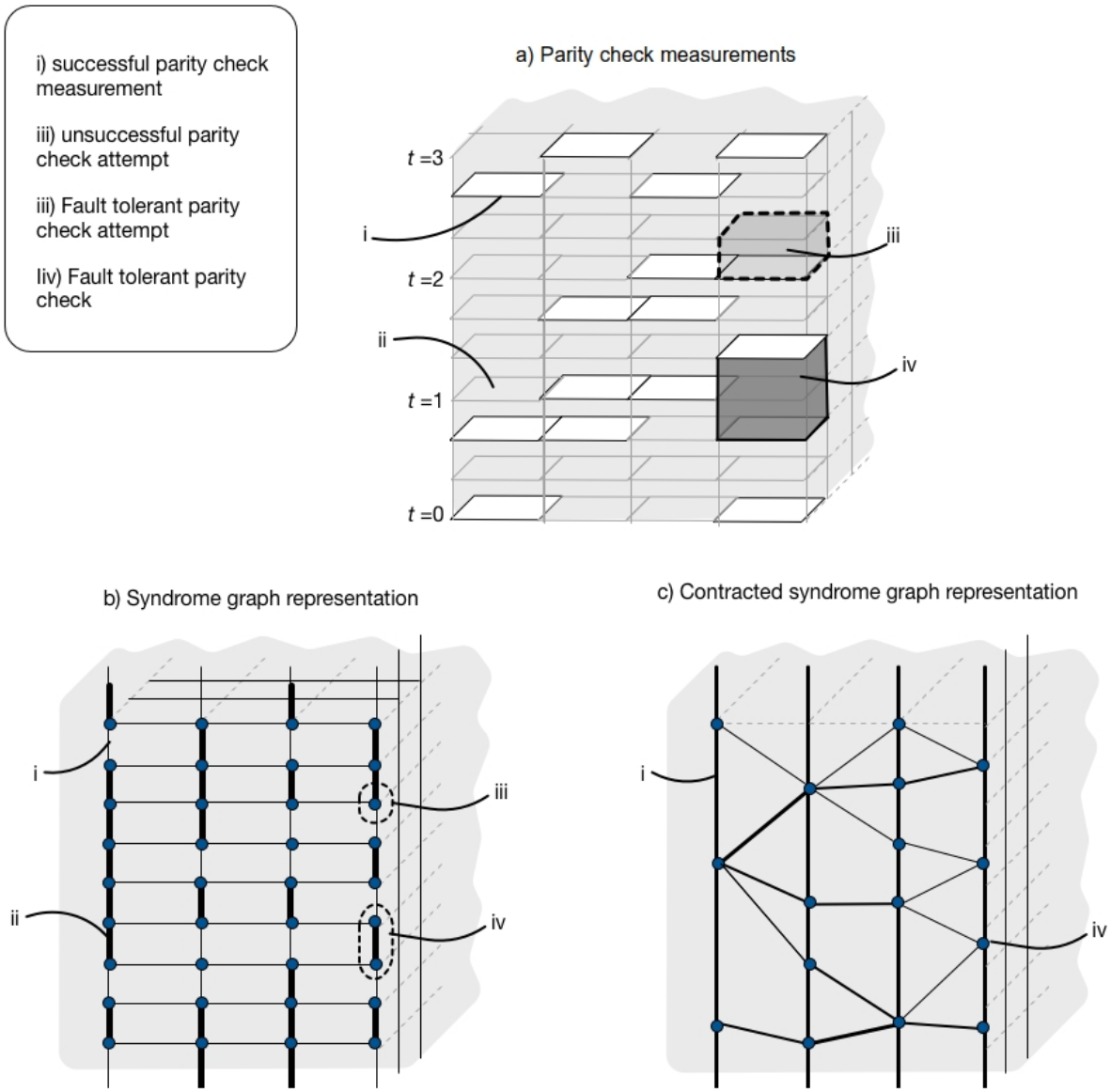


Figure 6.3: Constructing the contracted syndrome graph in the case of discrete asynchronous stabilizer measurement. We follow the method described first in [184]. (a) Collection of all parity check attempts through time, including successful and unsuccessful measurements. Two consecutive successful parity checks in time define a parity block. (b) Simple syndrome graph representation. Horizontal edges represent a possible physical Pauli error and vertical edges represent an attempted stabilizer measurement outcome. A vertical edge with unsuccessful parity check is marked as erased (bold edge). (c) The contracted syndrome graph representation. All vertices and vertical edges within a parity block are contracted into a single vertex. Horizontal edges connecting adjacent parity blocks are contracted into a single edge and new edge weights are calculated in order to reflect the degeneracy of the new contracted edges.

error occurs the two vertices connected to the corresponding edge will have their values flipped.

For a fault tolerance system there may be multiple possible syndrome graph representations that capture the same error model. We consider first the *simple syndrome graph* that is most naturally first derived from the parity check structure. We then consider the *contracted syndrome graph*.

6.2.2.1 Simple syndrome graph

When all parity measurements are performed synchronously the syndrome graph has a cubic structure. Time-like edges represent the possible measurement errors on parity checks, while space-like edges represent potential Pauli errors on the physical qubits. In our model of asynchronous measurement we have to modify this representation since not all parity checks return an outcome. This is done by marking an edge of the graph as erased when there is a corresponding measurement erasure. The net result is that multiple sequential erasures in time lead to the parity blocks of marked edges previously mentioned. The formulation of this system into a syndrome graph, named *simple syndrome graph*, is illustrated in Figure 6.3b. We note that the graph structure, i.e., its cubic structure, is the same in the every instance, the only difference being the position of the erased edges.

6.2.2.2 Contracted syndrome graph

Given a simple syndrome graph with a set of erased edges as shown in Figure 6.3b, we find an alternative representation without erased edges. When erasure is present, fault tolerant parity checks are only complete for each cluster of erased edges [184]. In our case this means simply treating all the vertices between two successful measurements as one vertex, i.e., considering a parity block as a vertex. By contracting the graph around the erased edges we arrive at the *contracted syndrome graph*. An example is shown in Figure 6.3c. The contraction resolves the problem of defining the anyons. The block anyons are identified as the anyons themselves.

Carrying out the contraction will often result in multi-edges in the graph, where two erased components were connected by multiple edges in the simple syndrome graph. These correspond to multiple possible errors that could cause the same syndrome. An equivalent representation that is more convenient for decoding is to instead represent these as a single edge with modified error probability. In general a pair of vertices connected by edges with given probabilities can be replaced by a single edge with the probability that an odd number of the edges have experienced an error.

For our scenario, as illustrated in Figure 6.3, this relates to the time-overlap of erased components in the simple syndrome graph. More formally, denote by p_Δ the probability that a qubit suffers an error between two consecutive parity check attempts. Call such probability p_Δ the *simulation error* (see Section 6.4 for more details on the simulations). Remember that the *physical error* is the probability that a qubit suffers an error per unit time, i.e., after $1/s$ parity

check attempts. The physical and simulation errors are related as follows: the probability that a qubit suffers an error after n measurement rounds equals the probability that during these n rounds its state is flipped an odd number of times (each with probability p_Δ), i.e.,

$$\sum_{m \text{ odd}}^{n} \binom{n}{m} p_\Delta^m (1-p_\Delta)^{n-m} = \frac{1}{2}\left(1-(1-2p_\Delta)^n\right), \tag{6.1}$$

where the equality can be seen from considering the binomial expansion of $((1-p)+p)^n + ((1-p)-p)^n$. Since a time unit represents $1/s$ measurement rounds on average, both quantities p and p_Δ are related via

$$p = \frac{1}{2}\left(1-(1-2p_\Delta)^{1/s}\right), \tag{6.2a}$$

$$p_\Delta = \frac{1}{2}\left(1-(1-2p)^{s}\right). \tag{6.2b}$$

The same reasoning applies when contracting the edges in the simple syndrome graph. The number of merged edges between two vertices i and j of the contracted syndrome graph is related to the time-overlap between their corresponding parity blocks. Let $\omega_{ij} = \min(t_i, t_j) - \max(t_{i-1}, t_{j-1})$ be the time overlap between two adjacent parity blocks (t_{i-1}, t_i) and (t_{j-1}, t_j). Therefore the number of merged edges is just ω_{ij}/s, where s is the synchronicity. The probability $\bar{p}(\omega_{ij})$ of a Pauli error occurring on the merged edge of the contracted syndrome graph is equal to the probability of a Pauli error occurring an odd number of times on the corresponding edges from the syndrome graph, which is given by Eq. (6.1), i.e.,

$$\bar{p}(\omega_{ij}) = \begin{cases} \frac{1}{2}\left(1-(1-2p_\Delta)^{\omega_{ij}/s}\right) & \text{(6.3a)} \\ \frac{1}{2}\left(1-(1-2p)^{\omega_{ij}}\right). & \text{(6.3b)} \end{cases}$$

Time-like (vertical) edges continue to represent possible measurement errors. The probability of a measurement error is still $q = p$.

6.2.2.3 Continuous stabilizer measurement

In the case of continuous stabilizer measurement when $s = 0$ there is no way to construct the simple syndrome graph. In this case we can build the contracted syndrome graph directly by sampling parity check measurement times. Given the locations of parity check measurements we directly construct the contracted syndrome graph by identifying a vertex with each parity block, and edges are placed between adjacent parity blocks with a non-zero time-overlap. Even though the concept of simulation error p_Δ is meaningless in the continuous measurement regime, the physical error p is still valid and thus, similarly to the discrete asynchronous case, the edges are weighted according to Eq. (6.3b).

6.3 Decoding

Given the syndrome graph (either simple or contracted), it is the job of the decoder to identify a *correction*. In our model this correction is a predicted set of flipped edges in the syndrome graph. An optimal decoder will identify a correction that minimizes the chance of producing a logical error. For practical use, efficient decoding algorithms [106, 66, 57] approximate the optimal solution whilst being computationally tractable.

6.3.1 Anyon pairing decoders

A correction in the surface code can be expressed as a matching of odd parity check vertices, i.e., anyons, since any error chain producing the same syndrome will have the same effect on the logical state. In this framework, as mentioned in Chapter 5, we can think about the problem as trying to match anyons in a way that maximizes the chance of success. This way of thinking is particularly amenable to working with the MWPM decoder [68, 52]. By defining a *matching graph* as a complete graph on the set of vertices identified as anyons and with the edge weight between two anyons as their corresponding distance in the syndrome graph (see Figure 5.7), the decoding problem is reduced to a minimum weight perfect matching problem.

When building the matching graph we would ideally like to compute the *anyon pairing probability*, the probability that *any* error created the observed syndrome. Let be $\mathbb{E}$ the set of all error chains between vertices i and j. The anyon pairing probability between anyons i and j is given by

$$P_{ij} = \sum_{E \in \mathbb{E}} P_E = P_0 + P_1 + P_2 + \dots, \tag{6.4}$$

where E is an error chain, P_E is its probability and we index all the error chains $E = 0, 1, 2, \dots$ from most likely to least likely.

6.3.2 Computing syndrome pairing probabilities

In practice computing the exact anyon pairing probability is a difficult problem, and we can instead compute an approximation to it. By looking at the structure of the full form of P_{ij} we can identify which terms it is appropriate to truncate in order to simplify the computational problem, without too damaging an effect on the threshold performance.

We consider an error model where each edge e in the syndrome graph represents an independent (but non-identical) error occurring with probability $p_e = \bar{p}(\omega_e) = \frac{1}{2}(1 - (1 - 2p)^{\omega_e})$ according to Eq. (6.3), where ω_e is the time-overlap associated with edge e. Using the

approximation $p_e \approx p\omega_e \ll 1$, Eq. (6.4) can be written as

$$P_{ij} = \sum_{E\in\mathbb{E}} \prod_{e\in E} p_e \prod_{e\notin E}(1-p_e) = C \sum_{E\in\mathbb{E}} \prod_{e\in E} \frac{p_e}{1-p_e} \tag{6.5}$$

$$\approx C \sum_{E\in\mathbb{E}} \prod_{e\in E} \frac{p\omega_e}{1-p\omega_e} \tag{6.6}$$

$$\approx C \sum_{E\in\mathbb{E}} \prod_{e\in E} (p\omega_e + (p\omega_e)^2) \tag{6.7}$$

$$= C \sum_{E\in\mathbb{E}} \left(\prod_{e\in E} p\omega_e\right) \left(\prod_{e\in E}(1+p\omega_e)\right) \tag{6.8}$$

$$\approx C \sum_{E\in\mathbb{E}} p^{|E|} \prod_{e\in E} \omega_e \tag{6.9}$$

$$= C \sum_{E\in\mathbb{E}} p^{|E|} \delta_E, \tag{6.10}$$

where $C = \prod_{\forall e}(1-p_e)$ is a constant for a given syndrome graph, $|E|$ is the length of the error chain E and we defined the quantity

$$\delta_E = \prod_{e\in E} \omega_e, \tag{6.11}$$

which is the product of the time-overlaps ω_e along the error chain. In Eq. (6.8) we approximated $\prod_{e\in E}(1+p\omega_e) \approx 1$. We expect time-overlaps ω_e to be smaller than 1 on average (since parity blocks have length 1 on average), so we can write $\prod_{e\in E}(1+p\omega_e) \lessapprox e^{p|E|}$. Thus, for small chains, the approximation $\prod_{e\in E}(1+p\omega_e) \approx 1$ is fairly accurate, but the same might not be true for large chains if the lattice size L is sufficiently large, since $e^{p|E|} \approx e^{pL}$ might be considerable if L is comparable to p^{-1}. The net effect is an underestimation of the actual probability of large chains. This, however, will cause little harm to our decoders since large error chains are rejected by MWPM-like decoders: the most-likely error configuration obtained from a perfect matching algorithm is, with high probability, made up of small error chains, whose probabilities are accurately approximated by the above equations.

We shall use the notation l_0 for the smallest number of vertices from an error chain connecting i and j, and $l_k = l_0 + k$ for $k = 1, 2, \ldots$. Let $\mathbb{E}_l$ be the set of error chains connecting i and j with l vertices. By grouping terms which have the same number of vertices we reach the expression for the pairing probability

$$P_{ij} \propto \sum_{l\geq l_0} \Omega_l \cdot p^l, \tag{6.12}$$

where we defined

$$\Omega_l := \sum_{E\in\mathbb{E}_l} \delta_E. \tag{6.13}$$

We will call this the *degeneracy term* – it is a factor that counts the number of paths with the same *number* of errors. More specifically, Ω_{l_k} is the $(k+1)$th-order degeneracy term. The value of P_{ij} itself is a feature of the physical system, but the terms P_E of the sum are dependent on the choice of syndrome graph. It is important to select a good representation such that the truncation of the sum provides a good approximation.

In the special case of full synchronous measurement ($s = 1$) with i.i.d. error model, $\delta_E = 1$ for all error chains E. Therefore, the first order degeneracy term Ω_{l_0} between two anyons i and j with coordinates (x_i, y_i, t_i) and (x_j, y_j, t_j) can be explicitly calculated as

$$\Omega_{l_0}(i,j) = \frac{(\Delta(x_i,x_j) + \Delta(y_i,y_j) + |t_i - t_j|)!}{(\Delta(x_i,x_j))!(\Delta(y_i,y_j))!(|t_i - t_j|)!}, \tag{6.14}$$

where $\Delta(x_i, x_j) = \min(x_i - x_j \bmod L, x_j - x_i \bmod L)$ is the horizontal distance on the lattice (and similarly for the vertical coordinates y). The above expression is simply the number of ways one can take $\Delta(x_i, x_j)$ steps in the x-direction, $\Delta(y_i, y_j)$ steps in the y-direction and $|t_i - t_j|$ steps in the t-direction. Such first order degeneracy term was previously considered by [184, 195, 30] in order to improve the performance of the MWPM decoder.

6.3.3 Decoding algorithms

We consider several decoders based on increasingly better approximations for P_{ij}, using both the simple and contracted syndrome graph. We first summarise these decoders before exploring them in more details.

1. **Unweighted Simple (US) decoder:** we approximate $\tilde{P}_{ij} \approx p^{l_0}$ in the *simple syndrome graph*. This is equivalent to considering the shortest path between two block anyons in the simple syndrome graph.

2. **Unweighted Contracted (UC) decoder:** we approximate $\tilde{P}_{ij} \approx p^{l_0}$ in the *contracted syndrome graph*. This takes into consideration the presence of multi-edges arising from multiple physical error configurations. Such approximation is equivalent to considering the shortest path between two anyons in the contracted syndrome graph.

3. **Weighted Contracted (WC) decoder:** we approximate $\tilde{P}_{ij} \approx P_0 \approx \delta_{E_0} \cdot p^{l_0}$, i.e., the most likely error chain in the contracted syndrome graph. This is equivalent to weighting each edge in the contracted syndrome graph by $\ln((1-p_e)/p_e) \approx \ln(1/p\omega_e))$ and finding the path with the minimum additive weight.

4. **Degenerate Weighted Contracted (DWC) decoder:** we approximate $\tilde{P}_{ij} \approx p^{l_0}(\Omega_{l_0} + p \cdot \Omega_{l_1})$, i.e., all error chains with the shortest and second shortest lengths.

5. **Average Position (AP) decoder:** as an alternative to the Unweighted Simple decoder, we position the anyons in the middle of their anyon blocks and proceed to reduce the

simple syndrome graph back to a cubic graph, as in a full synchronous regime with i.i.d. error model.

6.3.3.1 Unweighted decoding on simple syndrome graph

An initial approach is to approximate P_{ij} with the single most likely error configuration that could have caused this pair of syndromes, i.e., $\tilde{P}_{ij} = p^{l_0}$ (see Eq. (6.12)), within the simple syndrome graph in order to match the anyons and thus correct the code. We call such approach the *Unweighted Simple* (US) decoder. As previously mentioned, the anyons are not well defined in the simple syndrome graph, but a simple solution is to identify them with the anyon blocks. Since,

$$\max_E \ln \tilde{P}_{ij} = \text{const.} - \ln p^{-1} \min_E l_0, \tag{6.15}$$

the most likely error chain is the smallest one. Therefore, the weight between two anyons i and j into the matching graph will be distance $l_0(i,j)$ between their corresponding anyon blocks: given the cubic structure of the matching graph, the space distance between i and j is the Manhattan distance of their space coordinates, while their time distance is simply the vertical distance between both anyon blocks. If the blocks overlap in time, then the time distance is zero. More formally, let $(x_i, y_j, t_{i1}, t_{i2})$ and $(x_j, y_j, t_{j1}, t_{j2})$ be the coordinates of both anyon blocks. Then their distance, and thus the corresponding weight into the matching graph, is

$$w_{ij} = \Delta(x_i, x_j) + \Delta(y_i, y_j) + \max(t_{i1} - t_{j2}, t_{j1} - t_{i2}, 0), \tag{6.16}$$

where $\Delta(x_i, x_j)$ is the horizontal distance on the lattice (and similarly for the vertical coordinates y). Note that $\max(t_{i1} - t_{j2}, t_{j1} - t_{i2}, 0) = 0$ when the anyon blocks overlap in time.

6.3.3.2 Unweighted decoding on contracted syndrome graph

By moving to the contracted syndrome graph we can account for the degeneracy induced by erasure in weighted edges of the graph. We still use the approximation $\tilde{P}_{ij} \propto p^{l_0}$ and name the resulting decoder as *Unweighted Contracted* (UC) decoder. Eq. (6.15) is still valid, meaning that the weight assigned into the matching graph between two anyons i and j is their distance $l_0(i,j)$. However, there is no such close expression for $l_0(i,j)$ as Eq. (6.16) since the contracted syndrome graph does not have a regular structure. In general, the shortest paths in the *unweighted* contracted syndrome graph gives rise to a metric $d_{\bar{S}}$. The weight assignment is thus

$$w_{ij} = d_{\bar{S}}(i,j). \tag{6.17}$$

The metric $d_{\bar{S}}$ can be calculated using Dijkstra's algorithm [61], which we discuss further in Appendix 6.A.

6.3.3.3 Weighted decoding on contracted syndrome graph

The approximation $\tilde{P}_{ij} \propto p^{l_0}$, even though it captures multi-edges arising from erasure, ignores different edge weights from non-identical error probabilities. These can be taken into consideration by approximating $\tilde{P}_{ij} \approx \max_{E\in\mathbb{E}} P(E)$, i.e., the P_0 term in Eq. (6.4). This defines our *Weighted Contracted* (WC) decoder, where such approximation becomes

$$\max_E \ln P(E) = \text{const.} - \min_E \sum_{e\in E} \ln\left(\frac{1-p_e}{p_e}\right), \tag{6.18}$$

where the $p_e = \bar{p}(\omega_e)$ are given by Eq. (6.3). Thus finding the most likely error chain is equivalent to $\min_E \sum_{e\in E} \ln\left((1-p_e)/p_e\right)$. Hence, the WC decoder weights each edge in the contracted syndrome graph by $\ln((1-p_e)/p_e)$ and proceeds to find the path with the minimum additive weight. In other words, this weight assignment defines a metric d_S in the contracted syndrome graph.We stress that the WC decoder only makes the approximation $\tilde{P}_{ij} \approx P_0$. The term P_0 is calculated exactly and does not need to be approximated within the framework of Eqs. (6.5)-(6.10).

On the other hand, note that such metric can be approximated as $d_S(i,j) \approx l_0(i,j)\ln p^{-1} - \ln \delta_{E_0}$, i.e., as the shortest length of an error chain plus its parameter δ_{E_0} (see Eq. (6.11)). Therefore, the weight between two anyon blocks i and j is set as the length of the shortest path (within metric d_S) between them, i.e.,

$$w_{ij} = d_S(i,j) \approx l_0(i,j)\ln p^{-1} - \tau \ln \delta_{E_0}, \tag{6.19}$$

where we included a parameter τ, named *degeneracy parameter*, that can be tuned in order to improve the decoder performance.

6.3.3.4 Degenerated Weighted decoding on contracted syndrome graph

The WC decoder can be enhanced by keeping more terms in Eq. (6.4). From Eq. (6.12) we can keep the first two groups of terms with shortest lengths, i.e., $\mathbb{E}_{l_0}$ and $\mathbb{E}_{l_1}$. We call the corresponding degeneracy terms, Ω_{l_0} and Ω_{l_1}, *first* and *second* order degeneracy terms, respectively. Therefore the weight assignment between a pair of vertices i and j is, similarly to Eq. (6.19),

$$w_{ij} = l_0(i,j)\ln p^{-1} - \tau \ln\left(\Omega_{l_0} + p\cdot\Omega_{l_1}\right), \tag{6.20}$$

up to an additive constant, and where we again included the *degeneracy parameter* τ. The result is the *Degenerate Weighted Contracted* (DWC) decoder. Efficient computation of degeneracies Ω_{l_0} and Ω_{l_1} can be done via Dijkstra's algorithm, as explained in Appendix 6.A.

6.3.3.5 Average Position decoder

Even though Dijkstra's algorithm is linear in the number of edges, it might not be sufficiently fast for realistic quantum error correction. We then consider a simple and faster approximation

similar to the US decoder. The idea is to reduce the simple syndrome graph back to a cubic graph. Each anyon is identified at a time location in the middle of the corresponding anyon block. For a stabilizer operator with coordinates (x, y) and associated i-th anyon block defined by times t_{i-1} and t_i, the corresponding anyon has coordinates $(x, y, (t_i + t_{i-1})/2)$. We treat these syndromes as existing in a cubic syndrome graph, and compute the weight between syndromes using the Manhattan distance, i.e., given two anyons i and j with coordinates (x_i, y_i, t_i) and (x_j, y_j, t_j) (after placing them in the middle of the anyon blocks),

$$w_{ij} = \Delta(x_i, x_j) + \Delta(y_i, y_j) + w_{time}|t_i - t_j|, \tag{6.21}$$

where we introduce a tunable parameter w_{time}.

6.4 Simulation methods

We simulate each decoding across a range of varying measurement synchronicity using Monte Carlo sampling to sample error configurations, and applying a decoder to the error sample to determine whether a logical error was introduced. The simulations were all carried out in an $L \times L$ periodic lattice with $L \in \{10, 12, 14\}$ over a time interval T and repeated a number of $2 \cdot 10^5$ to 10^6 times. The simulations were carried out using the computational facilities of the Advanced Computing Research Centre, University of Bristol. The nodes had 16 and 64 cores, and computing a single threshold value would take between 15 hours for our simple decoders (Unweighted Simple and Average Position decoders) to 48 hours for our more complex decoders (Weighted Contracted decoder). All of our simulations used C++.

For simplicity we consider only phase-flip errors and X-type parity checks are considered. By symmetry the performance will be the same for bit-flip errors, which use Z-type parity checks. We have two distinct types of simulation, the discrete measurement regime, and continuous measurement.

6.4.1 Discrete measurement

Stabilizer measurements are made in discrete rounds, resulting in a simple syndrome graph with cubic structure. We perform N_s measurement rounds, thus generating a syndrome graph of size $L \times L \times N_s$, with $N_s = \lceil 2/s \rfloor L$, where where $\lceil x \rfloor$ denotes the closest integer to x. We assume periodic boundaries in space, and open boundaries in time, corresponding to initialization and destructive measurement of a toric code. The last measurement round is taken to be perfect in order to guarantee the existence of a perfect matching of the anyons.

At each measurement round we flip the value of each qubit with probability p_Δ, the *simulation error*, after which we perform the stabilizer measurements, each with probability s, the synchronicity. If an outcome is return, its value is flipped with probability q, the *measurement error*. The time scale is defined such that a stabilizer outcome per qubit is obtained at an

average rate of 1, i.e., after $1/s$ measurement rounds. Therefore the *physical error* p, i.e., the qubit error per unit time, is related to p_Δ according to Eq. (6.2). We always fix $q = p$.

The resulting simple syndrome graph with error configuration is then processed to construct the matching graph. Depending on the decoder this may first involve performing edge contraction of erased edges on the simple syndrome graph.

6.4.2 Continuous measurement

In the limit $s \to 0$ we cannot sample discretely and instead generate the contracted syndrome graph directly. We first note that the number of bit-flips that a qubit suffers in the discrete measurement regime is a Binomial distribution $B(N_s, p_\Delta)$. Thus, in the limit $s \to 0$ and $N_s \to \infty$, such Binomial distribution converges towards a *Poisson distribution* with parameter $\lim_{s\to 0} N_s \cdot p_\Delta$, according to the *Poisson limit theorem*. Using Eq. (6.2b) and that $N_s = T/s$, where T is time corresponding to the last measurement round, we obtain

$$\lim_{s\to 0} N_s \cdot p_\Delta = \frac{T}{2} \lim_{s\to 0} \frac{1-(1-2p)^s}{s} = \frac{T}{2} \ln\left(\frac{1}{1-2p}\right) \approx T \cdot p. \tag{6.22}$$

A similar reasoning applies to stabilizer measurements: the number of successful stabilizer measurements in the limit $s \to 0$ has a Poisson distribution with parameter $\lim_{s\to 0} s \cdot N_s = T$. The simulation for the continuous asynchronous regime ($s = 0$) is then performed by first setting a time interval T and a physical error p and then, for each qubit, sampling the number of bit-flips it suffers from a Poisson distribution with parameter $\frac{T}{2}\ln(1/(1-2p))$ and distributing these bit-flips uniformly at random along the time interval $(0, T)$. The same is done with the measurements: for each stabilizer operator, we sample a number of successful faulty measurements from a Poisson distribution with parameter T and distribute these measurements uniformly at random along the time interval $(0, T)$. We also include perfect measurements at time T to guarantee the existence of a perfect matching. A parity check is done by counting the number of errors of the adjacent qubits prior to the measurement time. If the number is even (odd), the measurement outcome is $+1$ (-1). For faulty measurements this outcome is flipped with probability $q = p$.

Given the locations of parity check measurements we then directly construct the contracted syndrome graph by identifying a vertex, v, with each successive pair of parity checks, and edges between neighboring check locations where fault tolerant parity checks have a non-zero time overlap, or, in other words, between adjacent parity blocks that overlap in time. We weight the edges with an error probability $p_e = \bar{p}(\omega_e)$, where ω_e is the time overlap of the parity blocks, as in Eq. (6.3), in case of the WC and DWC decoders.

6.4.3 Estimating the P_E terms

We also computed the average relative size between the first few P_E terms and P_0 from Eq. (6.4). The ratios were obtained by averaging over random pairs of anyons and contracted syndrome

graphs in a typical simulation as follows: given an $L \times L$ lattice, we first set a synchronicity s, a physical error p and a measurement error $q = p$. The value of p was chosen as the threshold of the WC decoder from Figure 6.5 at the given s (see next Section 6.5). The number of measurement rounds was set as $N_s = \lceil 2L/s \rfloor$ for $0 < s \leq 1$, and time interval $T = 2L$ for $s = 0$.[1] We then apply the usual procedure just described above of introducing physical errors and measuring the stabilizers to obtain a random contracted syndrome graph and a set of anyons. The ratio P_E/P_0 is averaged over all these anyon pairs. The final result was averaged over other random contracted syndrome graphs. The number of samples over random contracted syndrome graphs was set to $5000, 250, 20, 5, 2$ and 1 for lattice size L equal to $4, 6, 8, 10, 12$ and 14, respectively (if L is sufficiently large, then the number of samples can be small since one contracted syndrome graph has already enough anyon pairs). Computing the values P_E for each pair of anyons was performed via Yen's algorithm [210], which is a generalization of Dijkstra's algorithm for computing the k-shortest loopless paths in a graph with non-negative edge cost.[2]

6.5 Results

6.5.1 Full synchronous regime with i.i.d. error model

Figure 6.4 shows the effect of the first-order degeneracy term Ω_{l_0} from Eq. (6.14) on the threshold values for the full synchronous ($s = 1$) regime with faulty measurements and i.i.d. error model using the MWPM decoder. This dependence is measured by the degeneracy parameter τ (see Eq. (6.20) with only first-order degeneracy). The threshold value at $\tau = 0$ corresponds to the one of the usual MPMW decoder for the toric code, and was computed to be $2.937\% \pm 0.002\%$, in agreement with past results [193, 160, 188, 154, 95] (see also Figure 5.8b). Moreover, the dependence on τ is very similar to the one observed in [184, Figure 8] with perfect stabilizer measurements. In particular, the threshold does not peak at $\tau = 1$, as one would expect, but around $\tau = 1.4$, where the matching algorithm favours more degenerate paths, and by $\tau = 2$ it has already dropped significantly. The threshold at $\tau = 1$ is $3.050\% \pm 0.002\%$.

6.5.2 Asynchronous regime

Figure 6.5 shows the threshold dependence with synchronicity s for the five different decoders considered so far: the Unweighted Simple (US), Unweighted Contracted (UC), Weighted Contracted with degeneracy (DWC) and without degeneracy (WC), and Average Position (AP) decoders (the AP decoder was optimized by tuning the w_{time} parameter from Eq. (6.21)).

[1] For a fair comparison between $\langle P_i/P_0 \rangle$ for different values of s, the average number of vertices in the contracted syndrome graph needs to be the same, hence the $1/s$ factor for $s > 0$. The rounding operation, however, limits such comparison due to its non-linearity. Such limitation, nonetheless, is small with our parameters. Setting $N_s = \lceil \gamma L/s \rfloor$ for $\gamma > 2$ improves the comparison.

[2] Note that, if the length of the k-shortest path in the contracted syndrome graph is $l_k = \sum_e \ln((1-p_e)/p_e)$ (see Eq. (6.18)), then $P_k = \exp(-l_k)$.

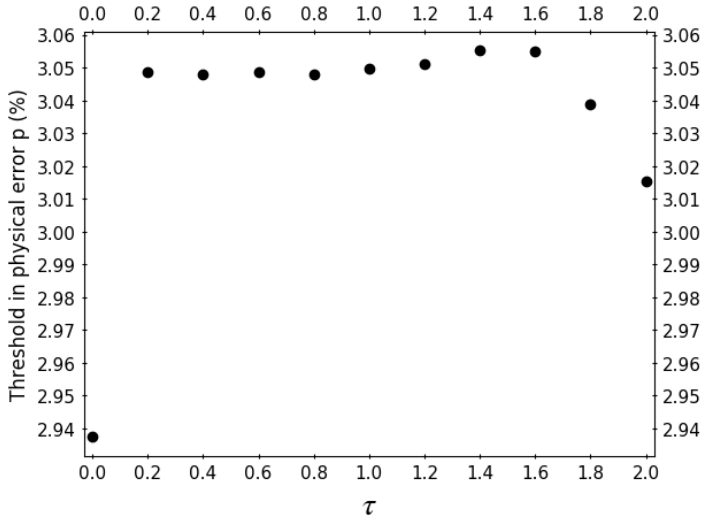
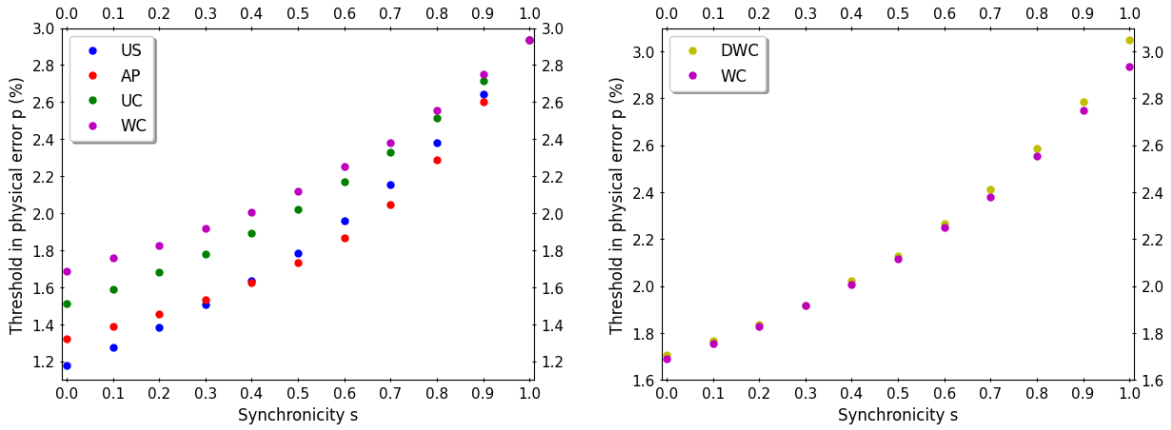


Figure 6.4: Threshold dependence with the degeneracy parameter τ in the full synchronous ($s = 1$) regime with faulty measurements and i.i.d. error model using the MWPM decoder. The thresholds at $\tau = 0$ and $\tau = 1$ are $2.937\% \pm 0.002\%$ and $3.050\% \pm 0.002\%$, respectively.

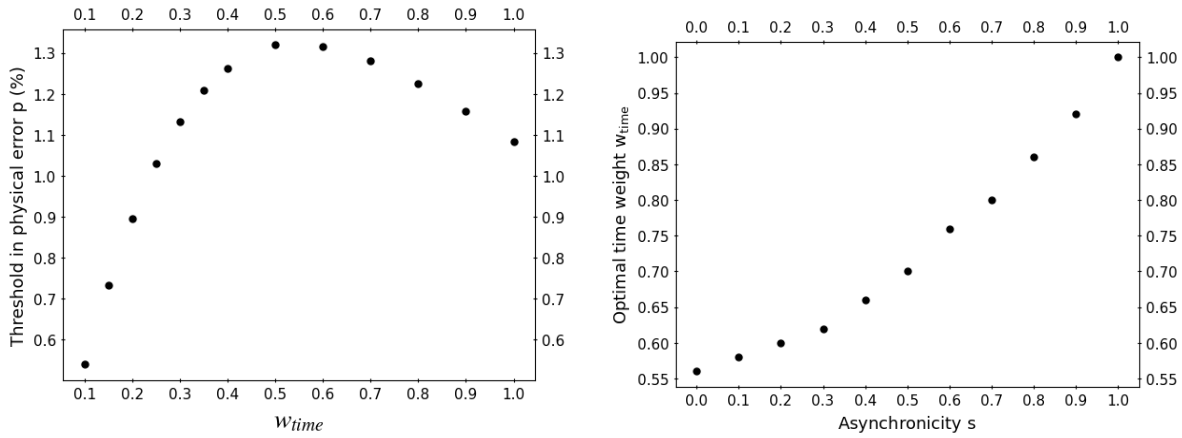


(a) (b)

Figure 6.5: Threshold dependence of all decoders with synchronicity s. (a) Comparison between Unweighted Simple (US), Unweighted Contracted (UC), Weighted Contracted without degeneracy (WC) and Average Position (AP) decoders. (b) Comparison between Weighted Contracted with degeneracy (DWC) and without degeneracy (WC) decoders.

Figure 6.5a compares all decoders but the DWC decoder. On the other hand, Figure 6.5b specifically compares the Weighed Contracted decoder with and without degeneracy. For $s = 1$ we have the usual MPMW decoder for the toric code with faulty measurements and i.i.d. error model, hence all decoders perform identically. As the synchronicity s decreases, the performance of all decoders decreases, as expected.

We can see that the inclusion of more features into the decoder in order to account for



(a) (b)

Figure 6.6: AP decoder dependence on the time weight w_{time}. (a) Threshold dependence on w_{time} in the continuous asynchronous ($s = 0$) regime with faulty measurements. (b) Optimal w_{time} as a function of synchronicity s. The optimal value at $s = 0$ is approximately 0.56.

asynchronism, e.g. syndrome graph contraction, edge weighting and degeneracy terms, gradually improves the threshold of the primitive US decoder. More interestingly, all the decoders can maintain a significant threshold value even at the limiting case when stabilizers are continuously measured in time ($s = 0$). Indeed, at $s = 0$, the threshold of $1.178\% \pm 0.001\%$ (US decoder) increases to $1.513\% \pm 0.002\%$ with contracting edges (UC decoder) and to $1.688\% \pm 0.001\%$ with also edge weighting (WC decoder). Moreover, the AP decoder, even though inferior to the US decoder for high values of s, outperforms it for high asynchronism, which is something that we currently have no explanation for.

6.5.3 AP decoder

The AP decoder in Figure 6.8a was optimized in terms of the time parameter w_{time} (see Eq. (6.21)). We explore its dependence on w_{time} in Figure 6.6. More specifically, Figure 6.6a shows the thresholds dependence on w_{time} in the full asynchronous scenario ($s = 0$), while Figure 6.6b shows the value of the optimal w_{time}, i.e., the value for which the threshold is maximum, as a function of s. The overall shape of Figure 6.6a is to be expected: both underestimation (large w_{time}) and overestimation (small w_{time}) of the time weight between anyons will worsen the performance of the AP decoder. In other words, for large w_{time}, anyons will be considered closer in time from what they really are, while, for small w_{time}, anyons will be considered farther. An optimal time weight w_{time} should be observed. Such point, in Figure 6.6a at $w_{time} \approx 0.56$, is very interesting, though, and we currently do not have arguments to explain it. Placing anyons in the middle of parity blocks seems to "shorten" the time distance, i.e., two anyons separated in space by some distance are equivalent to two anyons separated in time by roughly double the distance. In regards to Figure 6.6b, given that $w_{time} \approx 0.56$ at $s = 0$, we

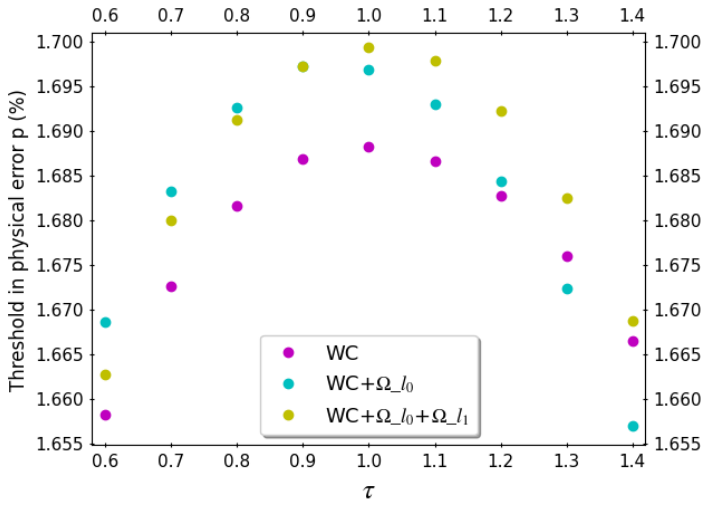


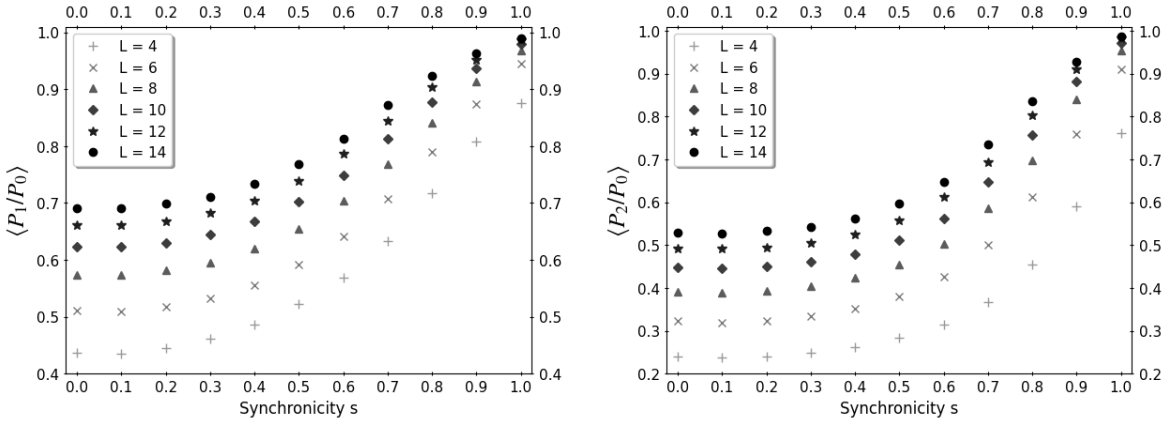
Figure 6.7: Threshold comparison between the WC decoder and its degeneracy versions with Ω_{l_0} and with both Ω_{l_0} and Ω_{l_1}. The comparison is with respect to the degeneracy parameter τ in the continuous asynchronous regime ($s = 0$).

expect a smooth interpolation between this value and the one of $w_{time} = 1$ at $s = 1$.

6.5.4 Degeneracy in the Weighted Contracted decoder

The improvement provided by the degeneracy terms Ω_{l_0} and Ω_{l_1} is quite small, as evident by Figure 6.5. We explore such improvement in more details for the continuous asynchronous regime. The WC decoder with degeneracy was defined by using $\tau \ln(\Omega_{l_0} + p \cdot \Omega_{l_1})$ (Eq. (6.20)) instead of $\tau \ln \delta_E$ (Eq. (6.19)). We could define an intermediary instance, where $\tau \ln \Omega_{l_0}$ is used instead of the full $\tau \ln(\Omega_{l_0} + p \cdot \Omega_{l_1})$. In Figure 6.7 we depict the improvement as a function of τ provided by including only the degeneracy term Ω_{l_0}, and by including both Ω_{l_0} and Ω_{l_1}, compared to just $\tau \ln \delta_E$. As expected, the introduction of more degeneracy improves the WC decoder threshold values. Moreover, the decoders' performances peak around $\tau = 1$, as expected given the discussion leading to Eq. (6.19): the probability P_0 considered by the WC decoder is best approximated by the case $\tau = 1$.

Something that stands out from Figures 6.5b and 6.7 is the fact that, while the introduction of high-order degeneracy like Ω_{l_1} does give higher threshold values compared to the base case of the WC decoder, this improvement becomes very small in the limit $s \to 0$. Even by $s = 0.9$ the reduction is significant. While at $s = 1$ the threshold increases from 2.937% to 3.050% (an $\sim 0.11\%$ additive improvement), at $s = 0$ it only increases from 1.688% to 1.699% (an $\sim 0.01\%$ additive improvement). This feature is not entirely surprising, given the following. If one assumes that the set of possible error probabilities p_e on each edge is very diverse, e.g. consider the case of continuous asynchronism where $p_e = \bar{p}(\omega_e)$ and ω_e can be any real number in $]0, T[$, then it becomes very unlikely to have two degenerate error chains. Therefore, for completely



(a) $\langle P_1/P_0 \rangle$ (b) $\langle P_2/P_0 \rangle$

Figure 6.8: Average ratios $\langle P_1/P_0 \rangle$ and $\langle P_2/P_0 \rangle$ as a function of synchronicity s for different sizes L of an $L \times L \times N_s$ lattice. Here $N_s = \lceil 2L/s \rfloor$ for $0 < s \leq 1$ and $N_0 = T = 2L$ for $s = 0$. The ratios were averaged over random contracted syndrome graphs given an $L \times L \times N_s$ lattice, a synchronicity s, a physical error p and a measurement error $q = p$. Given a synchronicity s, the value of p was chosen as the threshold of the WC decoder at s from Figure 6.5.

different error probabilities p_e, we expect most of the terms in Eq. (6.4) to be different from each other. This is in contrast to the fully synchronous regime ($s = 1$), where the first Ω_{l_0} terms are equal (Eq. (6.14)). Consequently, the leading term P_0 plays a more prominent role in the sum, and any truncation to it is less disruptive to its original value, when $s = 0$ compared to when $s = 1$.

In order to support the above claim, we shed some light on the relative size between the first P_E terms and P_0 which underlies the decrease in threshold values. Figures 6.8 and 6.9 numerically explores the average ratio between some of the first P_E terms and P_0 for different sizes of the toric code. More specifically, Figure 6.8a computes the average $\langle P_1/P_0 \rangle$ as a function of the synchronicity s in an $L \times L$ lattice for different sizes L. Figure 6.8b shows $\langle P_2/P_0 \rangle$ in a similar fashion. Turning to the results, in Figure 6.8 we see that for $s = 1$, the second and third terms in $\sum P_E$ are, on average, almost equal to P_0, specially for larger lattice sizes, meaning that it is very unlikely the first three terms to be different. Indeed, this happens when two out of three of the x, y and time distances ($\Delta(x_i, x_j)$, $\Delta(y_i, y_j)$ and $|t_i - t_j|$, respectively) are zero, so that $\Omega_{l_0} = 1$ (see Eq. (6.14)). On the other hand, for $s = 0$ we see that, on average, P_1 is $\sim 70\%$ of P_0, while P_2 is just $\sim 54\%$ (for $L = 14$). As a result, the leading term P_0 dominates the sum. Moreover, we notice two kinds of convergence: one with respect to s and another with respect to L. There is still room for increase regarding the convergence in terms of L, and it might be interesting to find these limits.

On the other hand, Figure 6.9 explores how much smaller the first few terms in $\sum P_E$ are in comparison with P_0 in the continuous asynchronous regime ($s = 0$). The average ratio $\langle P_i/P_0 \rangle$ is obtained for $i = 0, 1, \ldots, 10$. A clear exponential decay can be observed for each lattice size

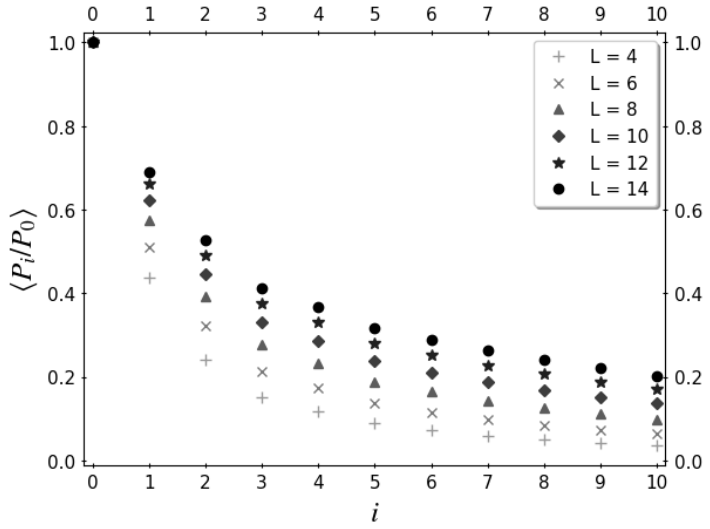


Figure 6.9: Average ratio $\langle P_i/P_0 \rangle$ as a function of i in the continuous asynchronous regime ($s = 0$) for different sizes L of an $L \times L \times 2L$ lattice. The ratios were averaged over random contracted syndrome graphs given an $L \times L \times 2L$ lattice, synchronicity $s = 0$, a physical error p and a measurement error $q = p$. We chose $p = 1.688\%$, the threshold of the WC decoder at $s = 0$ from Figure 6.5.

L. For $L = 14$ the P_{10} term is already 20% of the leading term P_0. The net result is again P_0 dominating the sum and decreasing the advantage provided by degeneracy terms.

6.6 Conclusions

The key point of this chapter was showing how asynchronism can be incorporated into the decoder while maintaining a high threshold. We started from the simple syndrome graph and contracted erased edges corresponding to unmeasured stabilizers, obtaining a graph we named contracted syndrome graph. Another explored aspect was the role of degeneracy terms when asynchronism is considered and how they could be included into the decoder. Based on the contracted syndrome graph and degeneracy terms, we defined a few decoders tailored to handle the asynchronism: the Weighted Contracted, Degenerate Weighted Contracted and the Average Position decoders. The WC decoder is a MWPM decoder, while DWC is an approximation that captures high-order degeneracy. On the other hand, the AP decoder works on a simplification of the problem in which the contracted syndrome graph is transformed back to a simple syndrome graph and a MWPM decoder is then used.

The performance of our decoders was explored by considering a simple error model where a stabilizer measurement outputs an outcome with some probability s which we called synchronicity. The limit $s = 0$ represents a continuous asynchronous regime where stabilizers are measured completely at random in time. The threshold values do decrease as the synchronicity tends

to zero, but a significant level can be maintained even under a completely continuous model of syndrome extraction, e.g. WC decoder with a threshold of 1.688% at $s = 0$. While our results were obtained with a simple error model, they show that erasure errors suffered by measurements can be efficiently handled by decoders. Studying the performance of the WC decoder under more realistic error models is a point to be considered in the future.

Regarding the role of degeneracy terms, the inclusion of the first and second order degeneracy terms Ω_{l_0} and Ω_{l_1} into the WC decoder produces strong evidence that such role becomes small in the continuous asynchronous regime. The DWC decoder with Ω_{l_0} and Ω_{l_1} perform almost identically to the WC decoder in the limit $s \to 0$, with a maximum threshold of 1.699%. This hints to the fact that in this limit the approximation done by a MWPM decoder of considering only the lowest weight error configuration becomes increasingly better. This was further backed up by our numerical results on the relative size between the most likely error configurations. We showed that, as the synchronicity decreases, the probability of the most likely error configuration becomes relatively higher than the probability of the subsequent ones. It might be interesting to understand this behaviour in a more qualitative manner with approximate analytical expressions.

If we ignore the DWC decoder given the little improvement on the threshold provided by first and second order degeneracy terms, the WC decoder achieves the highest thresholds over all proposed decoders for all levels of asynchronism. Moreover, it is relatively simple: by being a MPWM decoder, one only requires performing Dijkstra's algorithm on a rightly weighted graph (via Eq. (6.18)). However, even though the WC decoder might be considered efficient by being a polynomial-time algorithm, it could be too slow for practical applications. Indeed, running Dijkstra's algorithm requires time $O(|E| + |V| \log |E|)$, where $|E|$ and $|V|$ are the number of edges and vertices of the block syndrome graph. The block syndrome graph is fairly sparse, and it is reasonable to assume $|E| = O(|V|)$, meaning that each application of Dijkstra's algorithm takes $\tilde{O}(|V|) = \tilde{O}(L^3)$ time. Since Dijkstra's algorithm must be used once per anyon, this leads to the overall time complexity $\tilde{O}(L^6)$ (ignoring the running time of the blossom algorithm for finding the perfect matching of the anyons).

In order to remedy this, we proposed the simpler Average Position decoder, which skips any use of Dijkstra's algorithm by placing the anyons in the middle of the parity blocks and thus reduces finding a distance in a highly complex graph back to computing the Manhattan distance in a cubic graph, which can be done in constant time. The price is a decrease in threshold value down to 1.323% at $s = 0$, which is still reasonably high. Even though the overall time complexity is still $O(L^6)$, since the distance between the $O(L^6)$ anyon pairs must be computed, we found that, in practice, the AP decoder performed much faster than the WC decoder (and the UC decoder by extension, which is a simpler version of the WC decoder on unweighted graphs). Moreover, given the simple structure of the AP decoder, it might be possible to borrow previous techniques used to improve the basic MWPM decoder [125, 79] (some of these ideas could possibly be applied to the WC decoder as well). Finally, the AP decoder allows for

the introduction of auxiliary parameters like the time weight w_{time}, which must be tweaked depending on the error model. Understanding its performance as a function of w_{time} with more mathematical rigour is something that we did not tackle and should be considered in the future.

6.A Dijkstra's Algorithm

The weight between two anyons in the Weighted Contracted decoder is obtained via the metric d_S (Eq. (6.19)) defined from the contracted syndrome graph, which means that we are required to calculate shortest paths between two vertices in a graph. In order to do so efficiently, we used Dijkstra's algorithm [61]. Its run time is $O(|V|^2)$, where $|V|$ is the number of vertices in the graph. It is possible to improve its complexity to $O(|E| + |V| \log |V|)$, where $|E|$ is the number of edges, by replacing the min-priority queue from the original algorithm by a Fibonacci heap min-priority queue [80]. Here we used a binary min-priority heap, which is a heap data structure that takes the form of a binary tree, and a common variant of Dijkstra's algorithm that fixes a source vertex and calculate the shortest paths from it to all the other vertices in the graph, thus producing a shortest-path tree.

The algorithm works by initializing two values:

- `dist[]`, an array of distances from the `source` vertex to each vertex in the contracted syndrome graph. Initially, `dist[source]` $= 0$ and `dist[`u`]` $= \infty$ for all the other vertices u. As the algorithm progresses, the distance from `source` to each other vertex is updated.

- Q, a min-priority queue of unvisited vertices. Initially Q contains all the vertices. At the end of the algorithm, Q is empty.

A min-priority queue is an abstract data type with 3 basic operations: `add_with_priority()`, `decrease_priority()` and `extract_min()`. The operation `add_with_priority(`v`, dist[`v`])` adds v based on the value `dist[`v`]`. A min-priority queue will order its vertices v based on the increasing value of `dist[`v`]`. The `decrease_priority(`v`, dist[`v`])` updates the ordering according to a new value `dist[`v`]` of vertex v. And the operation `extract_min()` extracts the vertex with the minimum distance (located at the root of a binary or Fibonacci min-priority heap).

In summary, the algorithm first initialises the array of distances and the min-priority queue. Then, at each step, the vertex u with minimum `dist[`u`]` is extracted from Q and set as the `current` vertex. We consider all its children and calculate their tentative distances through `current`, i.e., `dist[current]` $+ d_S($`current`, `child`$)$. If `dist[child]` is greater than this tentative distance, then the distance of the child vertex is updated to the tentative value. We repeat this process until Q is empty.

We provide a pseudo-code for Dijkstra's algorithm.

Algorithm 1: Dijkstra's algorithm

Input: Contracted syndrome graph S and `source` vertex

```
create min-priority queue Q;
forall vertex v ∈ S do
    dist[v] ← INFINITY;
    Q.add_with_priority(v, dist[v]);
dist[source] ← 0;
Q.decrease_priority(source, dist[source]);
while Q ≠ ∅ do
    u ← Q.extract_min();
    for each neighbor v of u do
        aux ← dist[u] + d_S(u, v);
        if aux < dist[v] then
            dist[v] ← aux;
            Q.decrease_priority(v, dist[v]);
return dist[];
```

6.A.1 Degeneracy Terms

It is possible to use Dijkstra's algorithm in order to calculate the degeneracies Ω_{l_0} and Ω_{l_1} between the source vertex and all other vertices. This is done by using a simplified version of Dijkstra's algorithm for *unweighted* graphs. Similarly to the array of distances `dist[]`, two arrays of degeneracies are updated along the algorithm.

We initialize four values:

- l_0`[]`, an array of distances from `source` to each vertex in the *unweighted* contracted syndrome graph. Initially, l_0`[source]` $= 0$ and l_0`[`u`]` $= \infty$ for all the other vertices u. As the algorithm progresses, the distance from `source` to each other vertex is updated.

- Q, a queue of vertices to be explored. Initially Q contains only `source`. At the end of the algorithm, Q is empty.

- Ω_{l_0}`[]`, an array of first order degeneracies between `source` and each vertex in the contracted syndrome graph. Initially, Ω_{l_0}`[source]` $= 1$. As the algorithm progresses, Ω_{l_0}`[]` between `source` and each other vertex is updated.

- Ω_{l_1}`[]`, an array of second order degeneracies between `source` and each vertex in the contracted syndrome graph. Initially, Ω_{l_1}`[source]` $= 0$. As the algorithm progresses, Ω_{l_1}`[]` between `source` and each other vertex is updated.

Notice that Q here is a normal queue, and it provides only two operations: `add_end()` and `extract_first_element()`. The operation `add_end(`v`)` adds v at the end of the queue, and the operation `extract_first_element()` extracts the first element in the queue.

Dijkstra's algorithm for unweighted graphs is slightly different. This is because we need to update the distance of a given vertex only once. In summary, the algorithm first initialises the array of distances, the two arrays of degeneracies and the queue. Then, at each step, the first vertex of Q is extracted and set as the `current` vertex. We consider all its children. There are up to four cases we need to analyse:

1. l_0`[child]` $= \infty$: `child` has not been visited yet, so its distance is updated to l_0`[current]` $+ 1$. We also update Ω_{l_0}`[child]` as Ω_{l_0}`[current]` $\cdot\, \omega($`current, child`$)$, where $\omega(e)$ is the time overlap for edge e. The vertex `child` is then added to the end of Q.

2. l_0`[child]` $= l_0$`[current]` $+ 1$: `child` has been visited before through a different shortest path, then we update Ω_{l_0}`[child]` by adding Ω_{l_0}`[current]` $\cdot\, \omega($`current, child`$)$ to its old value.

3. l_0`[child]` $= l_0$`[current]`: `child` and `current` are equidistant from `source`, and the edge (`child`, `current`) belongs to a second shortest path, thus we need to update Ω_{l_1}`[child]` by adding Ω_{l_0}`[current]` $\cdot\, \omega($`current, child`$)$ to its old value.

4. l_0`[child]` $= l_0$`[current]` $- 1$: `current` has been visited before through a different second shortest path, then we update Ω_{l_1}`[current]` (and not Ω_{l_1}`[child]`) by adding Ω_{l_1}`[child]` $\cdot\, \omega($`current, child`$)$ to its old value.

We repeat the above procedure until Q is empty.

The update of Ω_{l_1}`[]` works by noticing that Dijkstra's algorithm explores the vertices in layers. First all vertices with distance 1 are queued and later explored, followed by all vertices with distance 2, and so on. A second shortest path can only happen if it is a combination of a shortest path with an edge between two vertices from the same layer, i.e., with the same distance l_0`[]`. Condition 3 (l_0`[child]` $= l_0$`[current]`) ensures that we go from a shortest path to a second shortest path via an edge between vertices in the same layer. We then need to use the first order degeneracy Ω_{l_0}`[]` to update the second order degeneracy Ω_{l_1}`[]`. This works because at this point of the algorithm all values of Ω_{l_0}`[]` for the given layer were already calculated. On the other hand, condition 4 (l_0`[child]` $= l_0$`[current]` $- 1$) ensures that we stay on a second shortest path, as the transition between shortest and second shortest paths happened in some previous layer. Hence the use of a second order degeneracy Ω_{l_1}`[]` to also update a second order degeneracy. This works since the values of Ω_{l_1}`[]` for a given layer are only calculated once all vertices from the layer are considered (differently from Ω_{l_0}`[]`, whose values are calculated when all the vertices from the *previous* layer are considered).

We provide a pseudo-code below for our adapted Dijkstra's algorithm. If we are not required to compute the second order degeneracy term $\Omega_{l_1}[]$, then all the lines regarding it can be ignored (lines $4, 18, 19, 20, 21$).

Algorithm 2: Dijkstra's algorithm for first and second order degeneracy

Input: Contracted syndrome graph S and `source` vertex

1 **forall** vertex $v \in S$ **do**
2 $\quad l_0[v] \leftarrow$ INFINITY;
3 $\quad \Omega_{l_0}[v] \leftarrow$ UNDEFINED;
4 $\quad \Omega_{l_1}[v] \leftarrow 0$;
5 $l_0[\texttt{source}] \leftarrow 0$;
6 $\Omega_{l_0}[\texttt{source}] \leftarrow 1$;
7 create queue Q;
8 Q.`add_end(source)`;
9 **while** $Q \neq \emptyset$ **do**
10 $\quad u \leftarrow Q$.`extract_first_element()`;
11 $\quad$ **for** each neighbor v of u **do**
12 $\quad\quad$ **if** $l_0[v] =$ INFINITY **then**
13 $\quad\quad\quad Q$.`add_end`(v);
14 $\quad\quad\quad l_0[v] = l_0[u] + 1$;
15 $\quad\quad\quad \Omega_{l_0}[v] = \Omega_{l_0}[u] \cdot \omega(u, v)$;
16 $\quad\quad$ **else if** $l_0[v] = l_0[u] + 1$ **then**
17 $\quad\quad\quad \Omega_{l_0}[v] \leftarrow \Omega_{l_0}[v] + \Omega_{l_0}[u] \cdot \omega(u, v)$;
18 $\quad\quad$ **else if** $l_0[v] = l_0[u]$ **then**
19 $\quad\quad\quad \Omega_{l_1}[v] \leftarrow \Omega_{l_1}[v] + \Omega_{l_0}[u] \cdot \omega(u, v)$;
20 $\quad\quad$ **else if** $l_0[v] = l_0[u] - 1$ **then**
21 $\quad\quad\quad \Omega_{l_1}[u] \leftarrow \Omega_{l_1}[u] + \Omega_{l_1}[v] \cdot \omega(u, v)$;
22 **return** $l_0[]$, $\Omega_{l_0}[]$, $\Omega_{l_1}[]$;

Bibliography

[1] S. Aaronson, *The learnability of quantum states*, Proceedings of The Royal Society of London. Series A. Mathematical, Physical and Engineering Sciences, 463 (2007), pp. 3089–3114. arXiv:quant-ph/0608142. 65

[2] S. Aaronson and A. Ambainis, *Forrelation: A problem that optimally separates quantum from classical computing*, SIAM Journal on Computing, 47 (2018), pp. 982–1038. arXiv:1411.5729. 43, 47

[3] S. Aaronson, A. Ambainis, J. Iraids, M. Kokainis, and J. Smotrovs, *Polynomials, quantum query complexity, and Grothendieck's inequality*, in 31st Conference on Computational Complexity, 2016. arXiv:1511.08682. 43, 47, 48

[4] E. A. Aguilar, M. Farkas, D. Martínez, M. Alvarado, J. Cariñe, G. B. Xavier, J. F. Barra, G. Cañas, M. Pawłowski, and G. Lima, *Certifying an irreducible 1024-dimensional photonic state using refined dimension witnesses*, Physical Review Letters, 120 (2018), p. 230503. arXiv:1710.04601. iii, 65

[5] D. Aharonov and M. Ben-Or, *Fault-tolerant quantum computation with constant error rate*, SIAM Journal on Computing, (2008). arXiv:quant-ph/9906129. 93

[6] D. Aharonov, V. Jones, and Z. Landau, *A polynomial quantum algorithm for approximating the Jones polynomial*, Algorithmica, 55 (2009), pp. 395–421. arXiv:quant-ph/0511096. 26

[7] J. Ahrens, P. Badziąg, M. Pawłowski, M. Żukowski, and M. Bourennane, *Experimental tests of classical and quantum dimensionality*, Physical Review Letters, 112 (2014), p. 140401. 65

[8] A. Ambainis, *Communication complexity in a 3-computer model*, Algorithmica, 16 (1996), pp. 298–301. 13, 23

[9] A. Ambainis, M. Banik, A. Chaturvedi, D. Kravchenko, and A. Rai, *Parity oblivious d-level random access codes and class of noncontextuality inequalities*, Quantum Information Processing, 18 (2019), p. 111. arXiv:1607.05490. 67

[10] A. Ambainis, D. Kravchenko, and A. Rai, *Optimal classical random access codes using single d-level systems*, arXiv preprint arXiv:1510.03045, (2015). 67

[11] A. Ambainis, D. Leung, L. Mancinska, and M. Ozols, *Quantum random access codes with shared randomness*, arXiv preprint arXiv:0810.2937, (2008). 66, 69, 75

[12] A. Ambainis, A. Nayak, A. Ta-Shma, and U. Vazirani, *Dense quantum coding and a lower bound for 1-way quantum automata*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing, 1999, pp. 376–383. `arXiv:quant-ph/9804043`. 3, 65, 66, 67, 69, 72, 75

[13] H. Anwar, B. J. Brown, E. T. Campbell, and D. E. Browne, *Fast decoders for qudit topological codes*, New Journal of Physics, 16 (2014), p. 063038. 95

[14] J. Arrazola and N. Lütkenhaus, *Quantum fingerprinting with coherent states and a constant mean number of photons*, Physical Review A, 89 (2014), p. 062305. `arXiv:1309.5005`. 21

[15] J. Aspnes, R. Beigel, M. Furst, and S. Rudich, *The expressive power of voting polynomials*, Combinatorica, 14 (1994), pp. 135–148. 50

[16] L. Babai and P. G. Kimmel, *Randomized simultaneous messages: solution of a problem of Yao in communication complexity*, in 12th Annual IEEE Conference on Computational Complexity (Ulm, 1997), IEEE Computer Soc., Los Alamitos, CA, 1997, pp. 239–246. 13

[17] C. Ballance, T. Harty, N. Linke, M. Sepiol, and D. Lucas, *High-fidelity quantum logic gates using trapped-ion hyperfine qubits*, Physical Review Letters, 117 (2016), p. 060504. 99

[18] H.-J. Bandelt and V. Chepoi, *Metric graph theory and geometry: A survey*, Discrete & Computational Geometry - DCG, 453 (2008). 22, 29

[19] Z. Bar-Yossef, T. Jayram, R. Krauthgamer, and R. Kumar, *Approximating edit distance efficiently*, in Proceedings of the 45th Annual Symposium Foundations of Computer Science, 2004, pp. 550–559. 14

[20] Z. Bar-Yossef, T. S. Jayram, and I. Kerenidis, *Exponential separation of quantum and classical one-way communication complexity*, in Proceedings of the 36th Annual ACM Symposium on Theory of Computing, ACM, New York, 2004, pp. 128–137. 13, 39, 40, 43

[21] J. Barnes and P. Hut, *A hierarchical $O(N \log N)$ force-calculation algorithm*, Nature, 324 (1986), pp. 446–449. 23

[22] J. N. de Beaudrap, *One-qubit fingerprinting schemes*, Physical Review A, 69 (2004), p. 022307. `arXiv:quant-ph/0309036`. 21

[23] W. Beckner, *Inequalities in Fourier analysis*, Ann. of Math., 102 (1975), pp. 159–182. 17

[24] A. Ben-Aroya, O. Regev, and R. de Wolf, *A hypercontractive inequality for matrix-valued functions with applications to quantum computing and LDCs*, in Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 2008, pp. 477–486. arXiv:0705.3806. 3, 19, 41, 45, 65, 67, 71

[25] P. Benioff, *The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines*, Journal of statistical physics, 22 (1980), pp. 563–591. 1

[26] ——, *Quantum mechanical hamiltonian models of turing machines*, Journal of Statistical Physics, 29 (1982), pp. 515–546. 1

[27] I. Benjamini, G. Kalai, and O. Schramm, *Noise sensitivity of Boolean functions and applications to percolation*, Publications Mathématiques de l'Institut des Hautes Études Scientifiques, 90 (1999), pp. 5–43. arXiv:math/9811157. 16

[28] C. H. Bennett and G. Brassard, *An update on quantum cryptography*, in Advances in cryptology (Santa Barbara, Calif., 1984), vol. 196 of Lecture Notes in Comput. Sci., Springer, Berlin, 1985, pp. 475–480. 1

[29] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, *Mixed-state entanglement and quantum error correction*, Physical Review A, 54 (1996), p. 3824. arXiv:quant-ph/9604024. 85

[30] M. E. Beverland, B. J. Brown, M. J. Kastoryano, and Q. Marolleau, *The role of entropy in topological quantum error correction*, Journal of Statistical Mechanics: Theory and Experiment, 2019 (2019), p. 073404. 100, 109

[31] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. M. Ziegler, *Oriented matroids*, vol. 46, Cambridge University Press, 1999. 29

[32] I. F. Blake and J. H. Gilchrist, *Addresses for graphs*, IEEE Trans. Inform. Theory, IT-19 (1973), pp. 683–688. 29

[33] A. Bonami, *Étude des coefficients Fourier des fonctions de $L^p(G)$*, in Annales de l'institut Fourier, vol. 20, 1970, pp. 335–402. 17

[34] C. Bonnington, S. Klavzar, and A. Lipovec, *On cubic and edge-critical isometric subgraphs of hypercubes*, Australasian J. Combinatorics, 28 (2003), pp. 217–224. 28, 30

[35] B. Brešar, W. Imrich, S. Klavžar, H. M. Mulder, and R. Škrekovski, *Tiled partial cubes*, Journal of Graph Theory, 40 (2002), pp. 91–103. 29

[36] D. Browne, *Topological codes and computation*, 2014. https://sites.google.com/site/danbrowneucl/teaching/lectures-on-topological-codes-and-quantum-computation. 85

[37] N. H. Bshouty, E. Mossel, R. O'Donnell, and R. A. Servedio, *Learning DNF from random walks*, J. Comput. System Sci., 71 (2005), pp. 250–265. 15

[38] H. Buhrman, R. Cleve, S. Massar, and R. de Wolf, *Nonlocality and communication complexity*, Reviews of modern physics, 82 (2010), p. 665. arXiv:0907.3584. 7

[39] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, *Quantum fingerprinting*, Physical Review Letters, 87 (2001), p. 167902. arXiv:quant-ph/0102001. 2, 14

[40] H. Buhrman, R. Cleve, and A. Wigderson, *Quantum vs. classical communication and computation*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, 1998, pp. 63–68. arXiv:quant-ph/9802040. 12

[41] H. Buhrman, N. Vereshchagin, and R. de. Wolf, *On computation and communication with small bias*, in Proceedings of the 22nd Annual IEEE Conference Computational Complexity, 2007, pp. 24–32. 43

[42] H. Buhrman and R. de Wolf, *Communication complexity lower bounds by polynomials*, in Proceedings of the 16th Annual IEEE Conference on Computational Complexity, IEEE, 2001, pp. 120–130. 65

[43] M. Bun and J. Thaler, *Dual lower bounds for approximate degree and markov–bernstein inequalities*, Information and Computation, 243 (2015), pp. 2–25. 44, 63

[44] C. Calabro, *The exponential complexity of satisfiability problems*, PhD thesis, UC San Diego, 2009. 72

[45] A. R. Calderbank and P. W. Shor, *Good quantum error-correcting codes exist*, Physical Review A, 54 (1996), p. 1098. arXiv:quant-ph/9512032. 85

[46] A. Casaccino, E. F. Galvão, and S. Severini, *Extrema of discrete Wigner functions and applications*, Physical Review A, 78 (2008), p. 022310. arXiv:0805.3466. 67

[47] A. Chailloux, I. Kerenidis, S. Kundu, and J. Sikora, *Optimal bounds for parity-oblivious random access codes*, New Journal of Physics, 18 (2016), p. 045003. arXiv:1404.5153. 67

[48] F. Chung, *Diameters of graphs: Old problems and new results*, Congressus Numerantium, 60 (1987), pp. 295–317. 32

[49] R. Clifford and T. Starikovskaya, *Approximate Hamming distance in a stream*, in 43rd International Colloquium on Automata, Languages, and Programming, Dagstuhl, Germany, 2016, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 20:1–20:14. arXiv:1602.07241. 24

[50] G. Cohen, *A nonconstructive upper bound on covering radius*, IEEE Transactions on Information Theory, 29 (1983), pp. 352–353. 66, 72

[51] R. Coleman, *On Krawtchouk polynomials*, arXiv preprint arXiv:1101.1798, (2011). 74

[52] W. Cook and A. Rohe, *Computing minimum-weight perfect matchings*, INFORMS journal on computing, 11 (1999), pp. 138–148. 107

[53] G. Cormode, M. Paterson, S. Sahinalp, and U. Vishkin, *Communication complexity of document exchange*, in Proceedings of the 11th ACM-SIAM Symposium Discrete Algorithms, 2000, pp. 197–206. 21

[54] B. Criger and I. Ashraf, *Multi-path summation for decoding 2d topological codes*, Quantum, 2 (2018), p. 102. 100

[55] W. van Dam, *Implausible consequences of superstrong nonlocality*, Natural Computing, 12 (2013), pp. 9–12. arXiv:quant-ph/0501159. 70, 78

[56] S. Dasgupta and A. Gupta, *An elementary proof of the Johnson-Lindenstrauss lemma*, International Computer Science Institute, Technical Report, 22 (1999), pp. 1–5. 30

[57] N. Delfosse and N. H. Nickerson, *Almost-linear time decoding algorithm for topological codes*, arXiv preprint arXiv:1709.06218, (2017). 107

[58] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, *Topological quantum memory*, Journal of Mathematical Physics, 43 (2002), pp. 4452–4505. arXiv:quant-ph/0110143. 85, 94, 100

[59] D. Deutsch and R. Jozsa, *Rapid solution of problems by quantum computation*, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences, 439 (1992), pp. 553–558. 1, 12

[60] M. Deza and M. Laurent, *Geometry of cuts and metrics*, in Algorithms and combinatorics, 1997. 29, 35

[61] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische mathematik, 1 (1959), pp. 269–271. 96, 110, 121

[62] D. Ž. Djoković, *Distance-preserving subgraphs of hypercubes*, Journal of Combinatorial Theory, Series B, 14 (1973), pp. 263–267. 29, 35, 38

[63] D. Dominici, *Asymptotic analysis of the Krawtchouk polynomials by the WKB method*, The Ramanujan Journal, 15 (2008), pp. 303–338. arXiv:math/0501042. 75

[64] J. Du, P. Zou, X. Peng, D. Oi, L. C. Kwek, C. H. Oh, and A. Ekert, *Experimental quantum multimeter and one-qubit fingerprinting*, Physical Review A, 74 (2006), p. 042319. 21

[65] D. P. Dubhashi and A. Panconesi, *Concentration of measure for the analysis of randomized algorithms*, Cambridge University Press, 2009. 25, 27, 47, 49

[66] G. Duclos-Cianci and D. Poulin, *Fast decoders for topological quantum codes*, Physical Review Letters, 104 (2010), p. 050504. arXiv:0911.0581. 95, 100, 107

[67] J. Edmonds, *Paths, trees, and flowers*, Canadian Journal of mathematics, 17 (1965), pp. 449–467. 95, 96

[68] ——, *Optimum branchings*, Journal of Research of the national Bureau of Standards B, 71 (1967), pp. 233–240. 99, 107

[69] P.-E. Emeriau, M. Howard, and S. Mansfield, *Quantum advantage in information retrieval*, arXiv preprint arXiv:2007.15643, (2020). iii, 67

[70] D. Eppstein, J.-C. Falmagne, and S. Ovchinnikov, *Media theory: interdisciplinary applied mathematics*, Springer Science & Business Media, 2007. 29

[71] J.-C. Falmagne, E. Cosyn, J.-P. Doignon, and N. Thiéry, *The assessment of knowledge, in theory and in practice*, in Formal concept analysis, Springer, 2006, pp. 61–79. 23

[72] M. Farkas, N. Guerrero, J. Cariñe, G. Cañas, and G. Lima, *Self-testing mutually unbiased bases in higher dimensions with space-division multiplexing optical fiber technology*, Physical Review Applied, 15 (2021), p. 014028. iii, 65

[73] M. Farkas and J. Kaniewski, *Self-testing mutually unbiased bases in the prepare-and-measure scenario*, Physical Review A, 99 (2019), p. 032316. arXiv:1803.00363. iii, 65, 67

[74] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. Wright, *Secure multiparty computation of approximations*, ACM Transactions on Algorithms, 2 (2006), pp. 435–472. 14

[75] R. P. Feynman, *Simulating physics with computers*, International Journal of Theoretical Physics, 21 (1982). 1

[76] ——, *Quantum mechanical computers*, Optics news, 11 (1985), pp. 11–20. 1

[77] W. M. Fitch and E. Margoliash, *Construction of phylogenetic trees*, Science, 155 (1967), pp. 279–284. 23

[78] G. Foletto, L. Calderaro, G. Vallone, and P. Villoresi, *Experimental demonstration of sequential quantum random access codes*, Physical Review Research, 2 (2020), p. 033205. 65

[79] A. G. Fowler, A. C. Whiteside, A. L. McInnes, and A. Rabbani, *Topological code autotune*, Physical Review X, 2 (2012), p. 041003. 120

[80] M. L. Fredman and R. E. Tarjan, *Fibonacci heaps and their uses in improved network optimization algorithms*, Journal of the ACM (JACM), 34 (1987), pp. 596–615. 121

[81] D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. de Wolf, *Exponential separations for one-way quantum communication complexity, with applications to cryptography*, in Proceedings of the 39th Annual ACM Symposium on Theory of Computing, ACM, New York, 2007, pp. 516–525. arXiv:quant-ph/0611209. 2, 13, 39, 40, 41, 44, 45, 50, 52

[82] D. Gavinsky, J. Kempe, O. Regev, and R. de Wolf, *Bounded-error quantum state identification and exponential separations in communication complexity*, SIAM Journal on Computing, 39 (2009), pp. 1–24. arXiv:quant-ph/0511013. 65

[83] D. Gavinsky, J. Kempe, and R. de. Wolf, *Quantum communication cannot simulate a public coin*, arXiv preprint quant-ph/0411051, (2004). 24

[84] ——, *Strengths and weaknesses of quantum fingerprinting*, in Proceedings of the 21st Annual IEEE Conference Computational Complexity, IEEE, 2006, pp. 288–298. arXiv:quant-ph/0603173. 15, 21, 24, 30

[85] M. Gimeno-Segovia, P. Shadbolt, D. E. Browne, and T. Rudolph, *From three-photon Greenberger-Horne-Zeilinger states to ballistic universal quantum computation*, Physical Review Letters, 115 (2015), p. 020502. 99

[86] D. Gottesman, *Class of quantum error-correcting codes saturating the quantum Hamming bound*, Physical Review A, 54 (1996), p. 1862. arXiv:quant-ph/9604038. 85

[87] ——, *The Heisenberg representation of quantum computers*, arXiv preprint quant-ph/9807006, (1998). 85

[88] ——, *An introduction to quantum error correction and fault-tolerant quantum computation*, in Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics, vol. 68, 2010, pp. 13–58. arXiv:0904.2557. 85

[89] R. L. Graham and H. O. Pollak, *On the addressing problem for loop switching*, The Bell System Technical Journal, 50 (1971), pp. 2495–2519. 22

[90] L. K. Grover, *A fast quantum mechanical algorithm for database search*, in Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 1996, pp. 212–219. arXiv:quant-ph/9605043. 1, 12

[91] A. Grudka, K. Horodecki, M. Horodecki, W. Kłobus, and M. Pawłowski, *When are Popescu-Rohrlich boxes and random access codes equivalent?*, Physical Review Letters, 113 (2014), p. 100401. arXiv:1307.7904. 65, 68

[92] A. Grudka, M. Horodecki, R. Horodecki, and A. Wójcik, *Nonsignaling quantum random access-code boxes*, Physical Review A, 92 (2015), p. 052312. arXiv:1403.1295. 68

[93] J.-Y. Guan, F. Xu, H.-L. Yin, Y. Li, W.-J. Zhang, S.-J. Chen, X.-Y. Yang, L. Li, L.-X. You, T.-Y. Chen, Z. Wang, Q. Zhang, and J.-W. Pan, *Observation of quantum fingerprinting beating the classical limit*, Physical Review Letters, 116 (2016), p. 240502. arXiv:1603.02089. 21

[94] A. Hameedi, D. Saha, P. Mironowicz, M. Pawłowski, and M. Bourennane, *Complementarity between entanglement-assisted and quantum distributed random access code*, Physical Review A, 95 (2017), p. 052345. arXiv:1701.08713. 65

[95] J. W. Harrington, *Analysis of quantum error-correcting codes: symplectic lattice codes and toric codes*, PhD thesis, California Institute of Technology, 2004. 96, 114

[96] M. Hayashi, K. Iwama, H. Nishimura, R. Raymond, and S. Yamashita, *(4, 1)-quantum random access coding does not exist—one qubit is not enough to recover one of four bits*, New Journal of Physics, 8 (2006), p. 129. arXiv:quant-ph/0604061. 65, 66

[97] ——, *Quantum network coding*, in Annual Symposium on Theoretical Aspects of Computer Science, Springer, 2007, pp. 610–621. arXiv:quant-ph/0601088. 65

[98] C. W. Helstrom, *Quantum detection and estimation theory*, Academic press, 1976. 44, 57, 79

[99] M. Herold, E. T. Campbell, J. Eisert, and M. J. Kastoryano, *Cellular-automaton decoders for topological quantum memories*, npj Quantum information, 1 (2015), p. 15010. arXiv:1406.2338. 95

[100] A. S. Holevo, *Bounds for the quantity of information transmitted by a quantum communication channel*, Problemy Peredachi Informatsii, 9 (1973), pp. 3–11. 1, 12

[101] A. Honecker, M. Picco, and P. Pujol, *Universality class of the Nishimori point in the $2D \pm J$ random-bond Ising model*, Physical Review Letters, 87 (2001), p. 047201. arXiv:cond-mat/0010143. 94

[102] T. Horiguchi and T. Morita, *Existence of the ferromagnetic phase in a random-bond Ising model on the square lattice*, Journal of Physics A: Mathematical and General, 15 (1982), p. L75. 94

[103] R. T. Horn, S. A. Babichev, K.-P. Marzlin, A. I. Lvovsky, and B. C. Sanders, *Single-qubit optical quantum fingerprinting*, Physical Review Letters, 95 (2005), p. 150502. arXiv:quant-ph/0410232. 21

[104] W. Huang, Y. Shi, S. Zhang, and Y. Zhu, *The communication complexity of the Hamming distance problem*, Information Processing Letters, 99 (2006), pp. 149–153. arXiv:quant-ph/0509181. 24

[105] B. D. Hughes, *Random walks and random environments: random walks*, vol. 1, Oxford University Press, 1995. 66

[106] A. Hutter, J. R. Wootton, and D. Loss, *Efficient Markov chain Monte Carlo algorithm for the surface code*, Physical Review A, 89 (2014), p. 022326. arXiv:1302.2669. 95, 107

[107] T. Imamichi and R. Raymond, *Constructions of quantum random access codes*, in Asian Quantum Information Symposium (AQIS), 2018. 66

[108] K. Iwama, H. Nishimura, R. Raymond, and S. Yamashita, *Unbounded-error one-way classical and quantum communication complexity*, in International Colloquium on Automata, Languages, and Programming, Springer, 2007, pp. 110–121. arXiv:0706.3265. 66

[109] J. C. Jackson, *An efficient membership-query algorithm for learning DNF with respect to the uniform distribution*, Journal of Computer and System Sciences, 55 (1997), pp. 414–440. 15

[110] W. Johnson and J. Lindenstrauss, *Extensions of Lipschitz mappings into a Hilbert space*, Contemporary mathematics, 26 (1984), p. 1. 30

[111] J. r. Justesen, *A class of constructive asymptotically good algebraic codes*, IEEE Trans. Inform. Theory, IT-18 (1972), pp. 652–656. 14

[112] J. Kahn, G. Kalai, and N. Linial, *The influence of variables on Boolean functions*, in Proceedings of the 29th Annual Symposium Foundations of Computer Science, IEEE, 1988, pp. 68–80. 18, 40

[113] Y. Kempner and V. E. Levit, *Geometry of poset antimatroids*, Electronic Notes in Discrete Mathematics, 40 (2013), pp. 169–173. 29

[114] I. Kerenidis, *Quantum encodings and applications to locally decodable codes and communication complexity*, University of California, Berkeley, 2004. 65

[115] I. Kerenidis and R. de Wolf, *Exponential lower bound for 2-query locally decodable codes via a quantum argument*, J. Comput. System Sci., 69 (2004), pp. 395–420. `arXiv:quant-ph/0208062`. 65

[116] J. Kim, P. Maunz, T. Kim, J. Hussman, R. Noek, A. Mehta, and C. Monroe, *Modular universal scalable ion-trap quantum computer (musiqc)*, in AIP Conference Proceedings, vol. 1363, American Institute of Physics, 2011, pp. 190–193. 99

[117] A. Y. Kitaev, *Quantum computations: algorithms and error correction*, Uspekhi Matematicheskikh Nauk, 52 (1997), pp. 53–112. 85

[118] ——, *Fault-tolerant quantum computation by anyons*, Annals of Physics, 303 (2003), pp. 2–30. `arXiv:quant-ph/9707021`. 3, 85

[119] H. Klauck, *On quantum and probabilistic communication: Las Vegas and one-way protocols*, in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, ACM, New York, 2000, pp. 644–651. iii, 32, 65, 67

[120] ——, *Lower bounds for quantum communication complexity*, in Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, IEEE, 2001, pp. 288–297. `arXiv:quant-ph/0106160`. 15

[121] ——, *One-way communication complexity and the Nečiporuk lower bound on formula size*, SIAM Journal on Computing, 37 (2007), pp. 552–583. `arXiv:cs/0111062`. iii, 67

[122] A. R. Klivans and R. A. Servedio, *Learning DNF in time* $2^{\tilde{O}(n^{1/3})}$, J. Comput. System Sci., 68 (2004), pp. 303–318. 15

[123] E. Knill, R. Laflamme, and G. J. Milburn, *A scheme for efficient quantum computation with linear optics*, nature, 409 (2001), pp. 46–52. 99

[124] E. Knill, R. Laflamme, and W. Zurek, *Threshold accuracy for quantum computation*, arXiv preprint quant-ph/9610011, (1996). 93

[125] V. Kolmogorov, *Blossom V: a new implementation of a minimum cost perfect matching algorithm*, Mathematical Programming Computation, 1 (2009), pp. 43–67. 99, 120

[126] I. Kremer, N. Nisan, and D. Ron, *On randomized one-round communication complexity*, in Proceedings of the 27th Annual ACM Symposium on Theory of Computing, 1995, pp. 596–605. 13, 22

[127] N. Kumar, E. Diamanti, and I. Kerenidis, *Efficient quantum communications with coherent state fingerprints over multiple channels*, Physical Review A, 95 (2017), p. 032337. 22

[128] E. Kushilevitz and Y. Mansour, *Learning decision trees using the Fourier spectrum*, SIAM Journal on Computing, 22 (1993), pp. 1331–1348. 15

[129] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge University Press, New York, NY, USA, 1997. 7, 10, 11, 12, 13

[130] E. Kushilevitz, R. Ostrovsky, and Y. Rabani, *Efficient search for approximate nearest neighbor in high dimensional spaces*, SIAM Journal on Computing, 30 (2000), pp. 457–474. 26

[131] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, *Perfect quantum error correcting code*, Physical Review Letters, 77 (1996), p. 198. 85

[132] H.-W. Li, Z.-Q. Yin, Y.-C. Wu, X.-B. Zou, S. Wang, W. Chen, G.-C. Guo, and Z.-F. Han, *Semi-device-independent random-number expansion without entanglement*, Physical Review A, 84 (2011), p. 034301. arXiv:1108.1480. 65

[133] O. Liabøtrø, *Improved classical and quantum random access codes*, Physical Review A, 95 (2017), p. 052315. arXiv:1607.02667. 66, 67

[134] N. Linial, E. London, and Y. Rabinovich, *The geometry of graphs and some of its algorithmic applications*, Combinatorica, 15 (1995), pp. 215–245. 34

[135] N. Linial, Y. Mansour, and N. Nisan, *Constant depth circuits, Fourier transform, and learnability*, J. Assoc. Comput. Mach., 40 (1993), pp. 607–620. 15

[136] Y. Liu and S. Zhang, *Quantum and randomized communication complexity of XOR functions in the SMP model*, in Electronic Colloquium on Computational Complexity (ECCC), vol. 20, 2013, p. 10. 24

[137] F. J. MacWilliams and N. J. A. Sloane, *The theory of error correcting codes*, vol. 16, Elsevier, 1977. 76

[138] L. Mančinska and S. A. Storgaard, *The geometry of Bloch space in the context of quantum random access codes*, arXiv preprint arXiv:2106.00155, (2021). 82

[139] Y. Manin, *Computable and uncomputable*, Sovetskoye Radio, Moscow, (1980). 1

[140] Y. Mansour, *An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution*, Journal of Computer and System Sciences, 50 (1995), pp. 543–550. 15

[141] S. Massar, *Quantum fingerprinting with a single particle*, Physical Review A, 71 (2005), p. 012310. arXiv:quant-ph/0305112. 21

[142] F. Merz and J. Chalker, *Two-dimensional random-bond Ising model, free fermions, and the network model*, Physical Review B, 65 (2002), p. 054425. arXiv:cond-mat/0106023. 94

[143] S. Micali and V. V. Vazirani, *An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs*, in Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science, IEEE, 1980, pp. 17–27. 96

[144] C. Monroe, R. Raussendorf, A. Ruthven, K. Brown, P. Maunz, L.-M. Duan, and J. Kim, *Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects*, Physical Review A, 89 (2014), p. 022317. 99

[145] A. Montanaro, *A new exponential separation between quantum and classical one-way communication complexity*, Quantum Inf. Comput., 11 (2011), pp. 574–591. arXiv:1007.3587. 40

[146] S. Morley-Short, S. Bartolucci, M. Gimeno-Segovia, P. Shadbolt, H. Cable, and T. Rudolph, *Physical-depth architectural requirements for generating universal photonic cluster states*, Quantum Science and Technology, 3 (2017), p. 015005. 99

[147] S. Morley-Short, M. Gimeno-Segovia, T. Rudolph, and H. Cable, *Loss-tolerant teleportation on large stabilizer states*, Quantum Science and Technology, 4 (2019), p. 025014. 99

[148] E. Mossel, R. O'Donnell, and R. A. Servedio, *Learning functions of k relevant variables*, J. Comput. System Sci., 69 (2004), pp. 421–434. 15

[149] S. Muhammad, A. Tavakoli, M. Kurant, M. Pawłowski, M. Żukowski, and M. Bourennane, *Quantum bidding in Bridge*, Physical Review X, 4 (2014), p. 021047. arXiv:1403.4280. 65

[150] A. Nayak, *Optimal lower bounds for quantum automata and random access codes*, in 40th Annual Symposium on Foundations of Computer Science (New York, 1999), IEEE Computer Soc., Los Alamitos, CA, 1999, pp. 369–376. arXiv:quant-ph/9904093. 65, 67

[151] J. v. Neumann, *Zur Theorie der Gesellschaftsspiele*, Mathematische annalen, 100 (1928), pp. 295–320. 11

[152] I. Newman, *Private vs. common random bits in communication complexity*, Information Processing Letters, 39 (1991), pp. 67–71. 10, 40, 43, 72

[153] I. Newman and M. Szegedy, *Public vs. private coin flips in one round communication games*, in Proceedings of the 38th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1996, pp. 561–570. 13

[154] N. Nickerson, *Practical fault-tolerant quantum computing*, PhD thesis, Imperial College London, 2015. 96, 99, 114

[155] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*, 2002. 85, 86

[156] R. O'Donnell, *Computational applications of noise sensitivity*, PhD thesis, Massachusetts Institute of Technology, 2003. 16

[157] ——, *Analysis of Boolean functions*, Cambridge University Press, 2014. 7, 15, 16, 17, 18, 82

[158] R. O'Donnell and R. A. Servedio, *Learning monotone decision trees in polynomial time*, SIAM Journal on Computing, 37 (2007), pp. 827–844. 15

[159] R. O'Donnell and Y. Zhao, *Polynomial bounds for decoupling, with applications*, in 31st Conference on Computational Complexity, vol. 50 of LIPIcs. Leibniz Int. Proc. Inform., Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016, pp. Art. No. 24, 18. arXiv:1512.01603. 47

[160] T. Ohno, G. Arakawa, I. Ichinose, and T. Matsui, *Phase structure of the random-plaquette Z_2 gauge model: accuracy threshold for a toric quantum memory*, Nuclear physics B, 697 (2004), pp. 462–480. arXiv:quant-ph/0401101. 94, 96, 114

[161] S. Ovchinnikov, *Partial cubes: structures, characterizations, and constructions*, Discrete Mathematics, 308 (2008), pp. 5597–5621. 28

[162] ——, *Graphs and cubes*, Springer Science & Business Media, 2011. 28

[163] K. Pang and A. El Gamal, *Communication complexity of computing the Hamming distance*, SIAM Journal on Computing, 15 (1986), pp. 932–947. 22

[164] M. Pawłowski and N. Brunner, *Semi-device-independent security of one-way quantum key distribution*, Physical Review A, 84 (2011), p. 010302. arXiv:1103.4105. 65

[165] M. Pawłowski, T. Paterek, D. Kaszlikowski, V. Scarani, A. Winter, and M. Żukowski, *Information causality as a physical principle*, Nature, 461 (2009), pp. 1101–1104. 65, 70, 72, 78, 79

[166] M. Pawłowski and M. Żukowski, *Entanglement-assisted random access codes*, Physical Review A, 81 (2010), p. 042326. arXiv:0906.0524. 67, 69, 70, 72, 78

[167] S. Pfalzner and P. Gibbon, *Many-body tree methods in physics*, Cambridge University Press, 2005. 23

[168] N. Polat, *Netlike partial cubes I. General properties*, Discrete Mathematics, 307 (2007), pp. 2704–2722. 29

[169] S. Popescu and D. Rohrlich, *Quantum nonlocality as an axiom*, Foundations of Physics, 24 (1994), pp. 379–385. 66, 69

[170] A. Rao and A. Yehudayoff, *Communication Complexity: and Applications*, Cambridge University Press, 2020. 7, 10, 11

[171] R. Raz, *Fourier analysis for probabilistic communication complexity*, Computational Complexity, 5 (1995), pp. 205–221. 15

[172] D. Saha and J. J. Borkała, *Multiparty quantum random access codes*, EPL (Europhysics Letters), 128 (2020), p. 30005. arXiv:1905.05668. 67

[173] B. Schumacher, *Quantum coding*, Physical Review A, 51 (1995), p. 2738. 1

[174] A. Scott, J. Walgate, and B. Sanders, *Optimal fingerprinting strategies with one-sided error*, Quantum Information & Computation, 7 (2007), pp. 243–264. arXiv:quant-ph/0507048. 21

[175] A. A. Sherstov, *Separating AC^0 from depth-2 majority circuits*, SIAM Journal on Computing, 38 (2009), pp. 2113–2129. 68

[176] ——, *The pattern matrix method*, SIAM Journal on Computing, 40 (2011), pp. 1969–2000. 44, 63, 68

[177] Y. Shi, X. Wu, and W. Yu, *Limits of quantum one-way communication by matrix hypercontractive inequality*, (2012). iii, 41, 44, 52, 56, 57, 58, 60

[178] P. W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, in Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Ieee, 1994, pp. 124–134. 1

[179] ——, *Scheme for reducing decoherence in quantum computer memory*, Physical Review A, 52 (1995), p. R2493. 1, 85

[180] ——, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal on Computing, 26 (1997), pp. 1484–1509. arXiv:quant-ph/9508027. 1

[181] S. V. Shpectorov, *On scale embeddings of graphs into hypercubes*, Eur. J. Comb., 14 (1993), pp. 117–130. 29, 35

[182] D. R. Simon, *On the power of quantum computation*, SIAM Journal on Computing, 26 (1997), pp. 1474–1483. 1

[183] R. W. Spekkens, D. H. Buzacott, A. J. Keehn, B. Toner, and G. J. Pryde, *Preparation contextuality powers parity-oblivious multiplexing*, Physical Review Letters, 102 (2009), p. 010401. arXiv:0805.1463. 65, 67

[184] T. M. Stace and S. D. Barrett, *Error correction and degeneracy in surface codes suffering loss*, Physical Review A, 81 (2010), p. 022317. arXiv:0912.1159. 3, 4, 100, 104, 105, 109, 114

[185] A. Steane, *Error correcting codes in quantum theory*, Physical Review Letters, 77 (1996), p. 793. 1, 85

[186] ——, *Multiple-particle interference and quantum error correction*, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 452 (1996), pp. 2551–2577. arXiv:quant-ph/9601029. 85

[187] G. V. Steeg and S. Wehner, *Relaxed uncertainty relations and information processing*, arXiv preprint arXiv:0811.3771, (2008). 68

[188] A. M. Stephens, *Fault-tolerant thresholds for quantum error correction with the surface code*, Physical Review A, 89 (2014), p. 022321. arXiv:1311.5003. 96, 114

[189] A. Tănăsescu, V.-F. Iliescu, and P. G. Popescu, *Optimal entanglement-assisted almost-random access codes*, Physical Review A, 101 (2020), p. 042309. 67, 70, 72, 78

[190] A. Tavakoli, A. Hameedi, B. Marques, and M. Bourennane, *Quantum random access codes using single d-level systems*, Physical Review Letters, 114 (2015), p. 170502. arXiv:1504.08105. iii, 65, 67

[191] A. Tavakoli, B. Marques, M. Pawłowski, and M. Bourennane, *Spatial versus sequential correlations for random access coding*, Physical Review A, 93 (2016), p. 032336. arXiv:1510.06277. 65

[192] E. Verbin and W. Yu, *The streaming complexity of cycle counting, sorting by reversals, and other problems*, in Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2011, pp. 11–25. 40, 44, 45, 52

[193] C. Wang, J. Harrington, and J. Preskill, *Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory*, Annals of Physics, 303 (2003), pp. 31–58. arXiv:quant-ph/0207088. 4, 95, 96, 100, 114

[194] X.-R. Wang, L.-Y. Wu, C.-X. Liu, T.-J. Liu, J. Li, and Q. Wang, *Experimental generation of entanglement-assisted quantum random access code*, Physical Review A, 99 (2019), p. 052313. 65

[195] F. H. Watson, H. Anwar, and D. E. Browne, *Fast fault-tolerant decoder for qubit and qudit surface codes*, Physical Review A, 92 (2015), p. 032309. arXiv:1411.3028. 95, 109

[196] S. Wehner, M. Christandl, and A. C. Doherty, *Lower bound on the dimension of a quantum system given measured data*, Physical Review A, 78 (2008), p. 062112. arXiv:0808.3960. 65

[197] S. Wehner and R. de Wolf, *Improved lower bounds for locally decodable codes and private information retrieval*, in Automata, languages and programming, vol. 3580 of Lecture Notes in Comput. Sci., Springer, Berlin, 2005, pp. 1424–1436. arXiv:quant-ph/0403140. 65

[198] S. Wiesner, *Conjugate coding*, ACM Sigact News, 15 (1983), pp. 78–88. 65

[199] A. Winter, *Quantum and classical message identification via quantum channels*, Festschrift "A S Holevo 60" (O. Hirota, ed.), (2004), pp. 171–188. arXiv:quant-ph/0401060. 21

[200] S. Wolf and J. Wullschleger, *Oblivious transfer and quantum non-locality*, in Proceedings of the International Symposium on Information Theory, 2005. ISIT 2005., IEEE, 2005, pp. 1745–1748. arXiv:quant-ph/0502030. 67

[201] R. de Wolf, *A brief introduction to Fourier analysis on the Boolean cube*, Theory of Computing, (2008), pp. 1–20. 7, 15, 18

[202] J. R. Wootton and D. Loss, *High threshold error correction for the surface code*, Physical Review Letters, 109 (2012), p. 160503. arXiv:1202.4316. 95

[203] A. Y. Wu, *Embedding of tree networks into hypercubes*, Journal of Parallel and Distributed Computing, 2 (1985), pp. 238–249. 28

[204] F. Xu, J. M. Arrazola, K. Wei, W. Wang, P. Palacios-Avila, C. Feng, S. Sajeed, N. Lütkenhaus, and H.-K. Lo, *Experimental quantum fingerprinting with weak coherent pulses*, Nature Communications, 6 (2015), p. 8735. arXiv:1503.05499. 21

[205] A. C. C. Yao, *Probabilistic computations: Toward a unified measure of complexity*, in Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 1977, pp. 222–227. 11, 52

[206] ——, *Some complexity questions related to distributive computing*, in Proceedings of the 11th Annual ACM Symposium on Theory of Computing, STOC '79, New York, NY, USA, 1979, ACM, pp. 209–213. 7, 13

[207] ——, *Lower bounds by probabilistic arguments*, in Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 1983, pp. 420–428. 11

[208] ——, *Quantum circuit complexity*, in Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 1993, pp. 352–361. 1, 11

[209] ——, *On the power of quantum fingerprinting*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, ACM, New York, 2003, pp. 77–81. 15, 21, 24, 25

[210] J. Y. Yen, *Finding the k shortest loopless paths in a network*, management Science, 17 (1971), pp. 712–716. 114

[211] S. Zhang, *On the power of lower bound methods for one-way quantum communication complexity*, in Proceedings of the 38th International Conference on Automata, Languages and Programming, 2011, pp. 49–60. 32, 34